

**VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského
inženýrství**

**Číslicový modul pro čtení rotačního kodéru
Digital module for Scanning the Rotary Knob**

2013

Petr Holub

Zadání bakalářské práce

Student:

Petr Holub

Studijní program:

B2649 Elektrotechnika

Studijní obor:

2601R004 Měřicí a řídicí technika

Téma:

**Číslicový modul pro čtení rotačního kodéru
Digital Module for Scanning the Rotary Knob**

Zásady pro vypracování:

1. Seznámení se s technikou FPGA a návrhem logických obvodů do programovatelných číslicových struktur.
2. Zpracování přehledu inkrementálních rotačních snímačů z pohledu výstupního signálu.
3. Návrh číslicové jednotky pro čtení rotačního kodéru typu P-RE20.
4. Implementace návrhu do Xilinx FPGA.
5. Experimentální ověření a zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

- [1] PINKER, Jiří a Martin POUPA. *Číslicové systémy a jazyk VHDL*. 1. vyd. Praha: BEN - technická literatura, 2006, 349 s. ISBN 80-730-0198-5. .
- [2] PARNELL, Karen a Nick MEHTA. *Programmable Logic Design Quick Start Handbook*. Xilinx Inc., 2003. 190s.
- [3] NOVÁK, Petr. Rotační inkrementální senzory. *Automa*, 2002, č.10, s. 32-33. ISSN 1210-9592.
- [4] XILINX. *Spartan-3A/3AN FPGA Starter Kit Board User Guide*. Xilinx, Inc., 2008. UG334 (v1.1) June 19, 2008. [cit. 13.11.2009]. Dostupné z: http://www.xilinx.com/support/documentation/boards_and_kits/ug334.pdf .

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Vladimír Kašík, Ph.D.**

Datum zadání: 18.11.2011

Datum odevzdání: 07.05.2013

doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Datum odevzdání 6.5.2013



Petr Holub

Abstrakt

Cílem této bakalářské práce je seznámit se s obvody FPGA, navrhnout a zrealizovat v programovatelném hradlovém poli číslicový obvod pro čtení pohybu rotačního kodéru typu P-RE20, který bude odolný vůči překmitům, vibracím a dalším chybám způsobeným při jeho provozování.

Abstract

The aim of this bachelor's thesis is to study the FPGA, to suggest and to implement a circuit digital module for scanning the Rotary Knob type P-RE20 in programmable gate array, which will be resistant to overshoots, vibrations and other errors caused during its operation.

Klíčová slova

FPGA, VHDL, Xilinx, inkrementální rotační kodér

Key Words

FPGA, VHDL, Xilinx, Incremental Rotary Knob

Seznam použitých symbolů a zkratek

ASIC	Application Specific Integrated Circuits
CLB	Configurable Logic Block
CPLD	Complex Programmable Logic Device
DCM	Digital Clock Manager
DDR	Double Data Rate
DLL	Delay-Locked Loop
FPGA	Field Programmable Gate Arrays
FSM	Finite State Machine
IOB	Input / Output Block
LUT	Look-Up Table
PI	Programmable Interconnect
PLD	Programmable Logic Device
RTL	Register Transfer Level
VHDL	Very high speed integrated circuit Hardware Description Language

Obsah

Úvod.....	2
1 Programovatelné hradlové pole	3
1.1 Historie	3
1.2 Architektura FPGA.....	3
1.3 Rozdělení FPGA podle uložení konfigurace	7
1.4 Návrh logických obvodů	9
2 Přehled inkrementálních rotačních snímačů	11
2.1 Úvod do inkrementálních rotačních snímačů	11
2.2 Výstupní obvody inkrementálních rotačních snímačů.....	13
2.3 Přehled inkrementálních rotačních snímačů.....	15
2.4 Rotační kodér P-RE20	21
3 Návrh číslicové jednotky pro čtení rotačního kodéru	22
3.1 Měření.....	22
3.2 Návrh	23
4 Implementace návrhu do Xilinx FPGA	27
4.1 Vzorkovací obvod s filtrem	27
4.2 Stavový automat	28
4.3 Ovládání čítače a asynchronní reset obvodu	30
4.4 Konfigurace a universálnost použití číslicového modulu.....	31
5 Experimentální ověření a zhodnocení dosažených výsledků.....	32
5.1 Ověření funkčnosti dekódování kvadraturních signálů v simulaci.....	32
5.2 Praktické ověření funkčnosti reálného číslicového modulu	34
6 Závěr.....	37
7 Literatura.....	38
8 Přílohy.....	40

Úvod

Rotační kodéry jsou v dnešní době čím dál víc součástí jak našeho osobního života, tak techniky v automatizaci. Aniž bychom si to uvědomovali, používáme je v radiích, monitorech a jiných zařízeních k jejich ovládání, protože společně s integrovaným tlačítkem jsou ideálními ovládacími prvky umožňující ovládání i složitých menu. V automatizaci jsou zase nedílnou součástí pro zpětnovazební řízení rotačních pohonů, kde je důležité přesné zjištění pohybů těchto zařízení. A právě bezchybné čtení a správný návrh obvodu je to, čím se tato práce zabývá. První část je věnována seznámení se s obvody FPGA jako jedním z možných obvodů pro zpracování kvadratických signálů vycházejících z inkrementálních rotačních kodérů. Druhá kapitola je přímo zaměřena na inkrementální rotační snímače i s přehledem dostupných typů na trhu. Dále se tato práce zabývá návrhem obvodu pro čtení konkrétního mechanického inkrementálního rotačního kodéru typu P-RE20, při kterém je třeba odstranit chyby vznikající při provozování rotačních kodérů, mezi které patří zákmity vznikající při spínání a rozpínání mechanických kontaktů, což se řeší parametrizovatelným číslicovým filtrem a zvolením vhodné vzorkovací frekvence. Čtvrtá kapitola popisuje implementaci navrženého obvodu do programovatelného hradlového pole Spartan 3AN XC3S700AN pomocí programu ISE Design Suite od firmy Xilinx. V tomto programu je implementován simulační program ISim, v kterém byl nakonfigurovaný obvod otestován a následně ověřen i v reálných zkouškách.

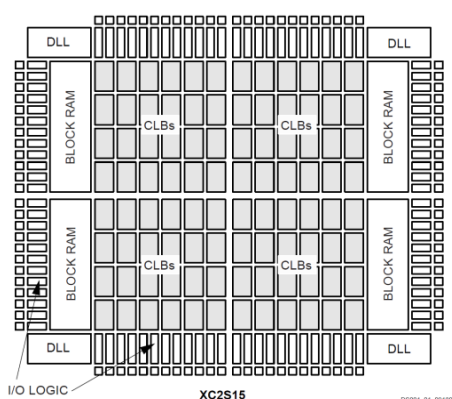
1 Programovatelné hradlové pole

FPGA jsou nejsložitější speciální integrované obvody z PLD obvodů, tvořené různě složitými programovatelnými bloky, které jsou navzájem propojeny konfigurovatelnou maticí spojů. Programovatelnost neboli konfigurovatelnost je zásadní vlastnost odlišující tyto obvody od zákaznických integrovaných obvodů ASIC.

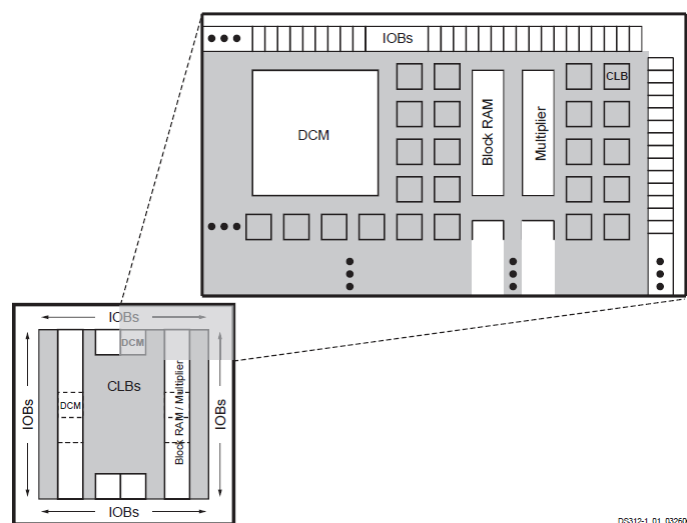
1.1 Historie

V osmdesátých letech se vyráběly menší programovatelné obvody typu PAL (Programmable Array Logic), vycházející z bipolárních pamětí PROM, v kterých bylo umožněno realizovat do deseti logických funkcí včetně případných registrů a zpětných vazeb. Tyto obvody byly jednorázově programovatelné s poměrně rychlými obvody. Postupně se pro tyto i složitější obvody vžila zkratka PLD. Později se začala v programovatelných obvodech používat technologie EPROM, založená na technologii CMOS, která umožňovala mazání a znovunaprogramovatelnost. Nové technologie umožňující větší hustotu obvodu na čipu a tím složitější obvody, daly vznik CPLD, což je vlastně obdoba několika obvodů PLD v jednom pouzdře doplněná programovatelným propojením těchto obvodů. Větší aplikace, které svým obsahem přesáhly kapacitu obvodů CPLD byly provedeny technologií hradlových polí GA (Gate Array), které vyžadovaly výrobu několika masek pro každý typ obvodu. Tato hradlová pole měla za sebou několik let vývoje a užívání, ve kterých bylo možno ověřit různá uspořádání logických obvodů a propojovacích prostředků. Vzhledem k velikosti realizovaného obvodu, bylo třeba respektovat vlastnosti prostředků CAD pro podporu návrhu v počítačích. Tyto zkušenosti využila firma Xilinx, která vyvinula v roce 1985 první zákaznický programovatelné hradlové pole, nám už známe pod označením FPGA. Základní blok obvodu byl tvořen jen jedním klopným obvodem a jedním programovatelným kombinačním členem, což bylo mnohem méně než u obvodu PLD.

1.2 Architektura FPGA



Obrázek 1: Architektura čipu generace Spartan II [10]

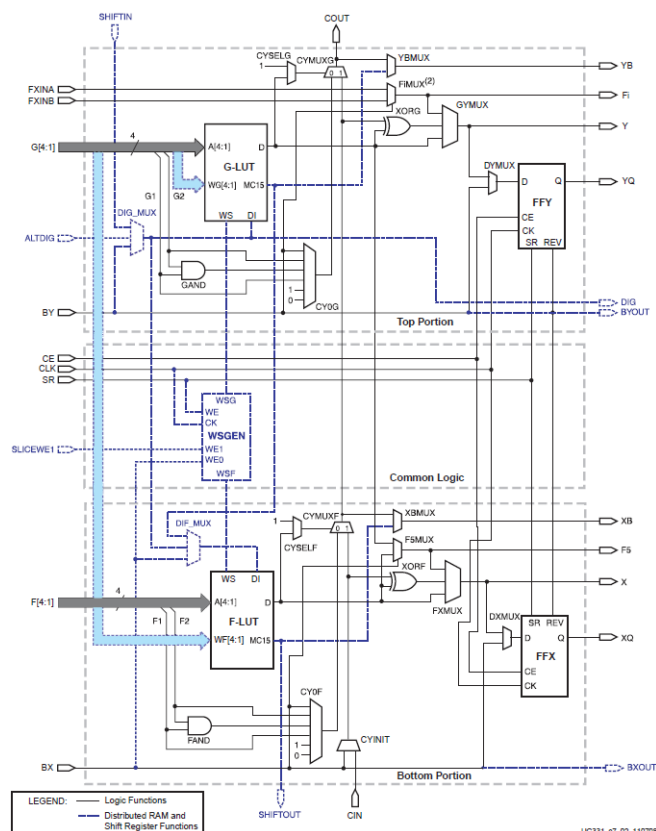


Obrázek 2: Architektura čipu generace Spartan 3 [11]

Základní vnitřní architektura FPGA obvodů je tvořena programovatelnou maticí konfigurovatelných logických bloků CLB, několika rovnoměrně rozvrženými bloky paměti RAM a modulů DLL nebo DCM. Tyto pole bloků jsou po obvodu obklopeny vstupně výstupními bloky IOB, a to vše je propojeno programovatelnou propojovací strukturou PI. Dnešní architektura FPGA je ale samozřejmě složitější, najdeme zde generátory hodin, podpůrné obvody pro aritmetické operace, procesorová jádra, hardwarové násobičky, bloky pro podporu sériové komunikace, AD převodníky, bloky pro řízení odběru atd.

Configurable Logic Block

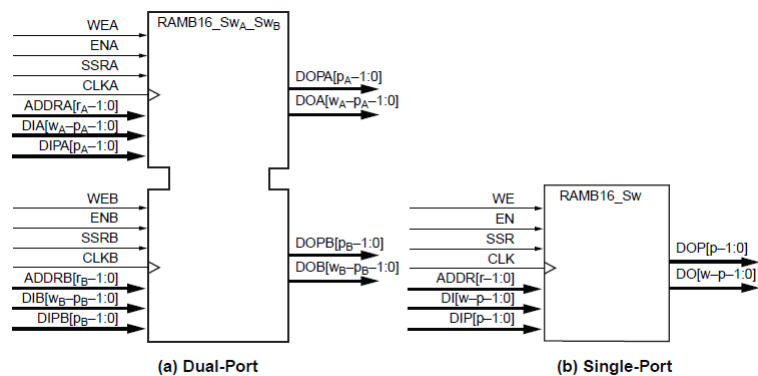
CLB můžeme přirovnat k malým blokům CPLD, a jedná se o hlavní nástroj pro provádění sekvenční i kombinační logiky. U čipu generace Spartan 3 je jich od 176 (u typu XC3S50A/AN) do 8320 (u typu XC3S5000) a jsou tvořeny čtyřmi identickými řezy. Každý řez, jak je znázorněno na obrázku č. 3, obsahuje dvě Look-Up Tables (LUT), dva klopné obvody typu D a vše je pospojováno uvnitř CLB programovatelnými multiplexory. Pro implementaci kombinačních obvodů slouží LUT - je to vlastně paměť, kterou lze realizovat všechny logické funkce čtyř proměnných s jedním výstupem. Pomocí multiplexorů můžeme spojovat LUT a vytvářet funkce více proměnných. LUT můžeme použít také jako 16 x 1 bitovou paměť RAM, nebo jako 16-bitový posuvný registr, který můžeme zase pomocí multiplexorů spojovat. Ať už vytvoříme jakoukoliv funkci v LUT, tak je velkou předností v FPGA vlastnost, že všechny tyto funkce budou mít stejné zpoždění signálu. Pro implementaci sekvenční logiky slouží konfigurovatelný klopný obvod typu D, který může být jak citlivý na náběžnou nebo sestupnou hranu hodin (D flip flop) tak na úroveň (D latch). V CLB je také umožněno propojit pomocí CIN a COUT některé signály logických bloků přímo se sousedními CLB bez nutnosti využívat globální propojovací matici, čímž docílíme spojení s mnohem menším zpožděním a umožníme tak realizovat rychlé obvody šíření přenosu nezbytné pro sčítačky nebo násobičky.



Obrázek 3: Zjednodušené schéma jednoho řezu CLB [11]

Block RAM

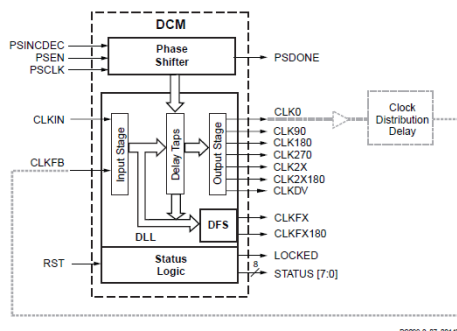
Rychlá statická paměť RAM je u generace Spartan 3 rozdělená do bloků o velikosti 18432 bitů, které jsou uspořádány na čipu do sloupců. U jednotlivých modelů je pak od 3 až do 104 bloků této paměti RAM, kde u každého bloku je možnost vybrat z několika režimů organizace paměti, například 16Kx1, 4Kx4, 1Kx16(bez parity), až po 256x72, kdy u všech režimů umožňuje full-speed provoz na frekvenci větší než 250 MHz. Každý blok paměti má dva nezávislé přístupové porty nazvané Port A a Port B, které jsou plně symetrické a vzájemně zaměnitelné.



Obrázek 4: Bloky paměti RAM v provedení (a) Dual-Port a (b) Single-Port [11]

Digital Clock Manager

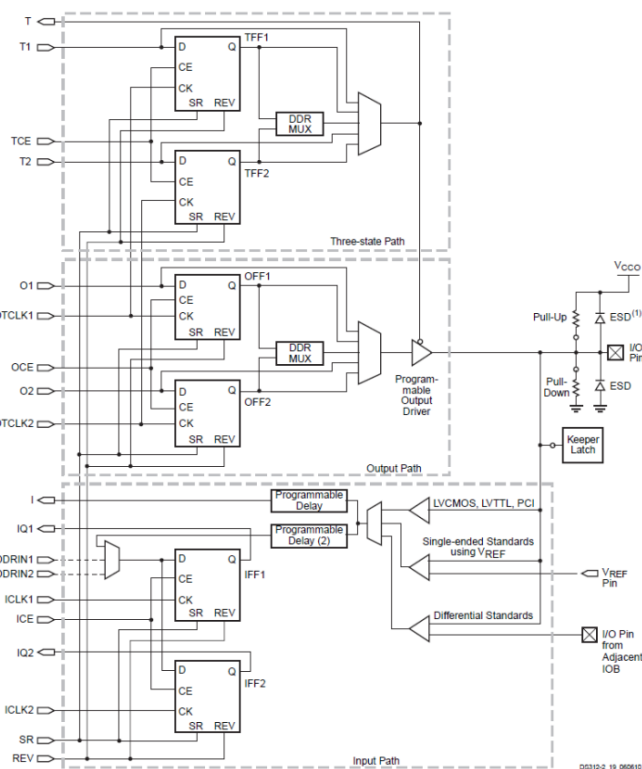
Slouží k úpravě hodinových signálů - srovnává například v celém obvodu zpoždění externích hodinových signálů přivedených na vstup FPGA, umožňuje dělit i násobit frekvenci signálů a provádí filtraci kmitočtového neklidu.



Obrázek 5: Funkční blokové schéma DCM [11]

Input Output Block

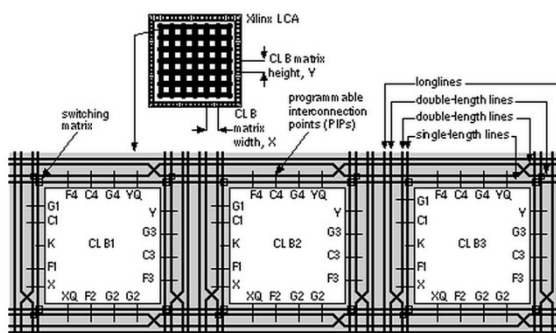
Umožňuje propojení FPGA s okolním, každý blok může pracovat ve třech režimech: vstupní, výstupní nebo obousměrný. U čipu generace Spartan 3 je v závislosti na typu modelu od 108 do 633 IOBs. Navíc jsou u této generace zdvojené klopné obvody typu D pro dvounásobně rychlejší přenos dat použitím techniky DDR. IOB podporuje více než 20 standardů rozhraní s různými specifikacemi pro proud a napětí I/O signálů a obsahuje tři hlavní signálové cesty: třístavovou, výstupní a vstupní, jak je vidět na obrázku č. 6.



Obrázek 6: Zjednodušené schéma IOB [11]

Programmable Interconnect

Je programovatelná propojovací struktura určená k propojení všech uvedených bloků. Struktura PI je u generace Spartan 3 tvořena čtyřmi druhy propojení: long lines, hex lines, double lines a direct lines. Long lines jsou 24 vodičové linky, které jsou umístěny horizontálně i vertikálně a spojují v jednom bodě šest bloků. Hex lines jsou 8 vodičové linky, které jsou umístěny horizontálně i vertikálně a spojují každý třetí blok. Double lines jsou 8 vodičové linky umožňující spojení s každým jiným blokem v čipu, a to jak v horizontálním tak i vertikálním směru a jsou flexibilnější v porovnání s long lines a hex lines. Direct lines jsou linky spojující sousední bloky jak horizontálně a vertikálně tak i diagonálně.



Obrázek 7: Ukázka PI u Xilinx LCA [12]

1.3 Rozdělení FPGA podle uložení konfigurace

U FPGA rozlišujeme dva druhy uložení konfigurace hradlového pole - s volatilní konfigurací a s nevolatilní konfigurací. Oba tyto způsoby přinášejí své výhody i nevýhody, které výrazně ovlivňují rozhodování při výběru těchto technologií pro novou aplikaci FPGA obvodů.

„FPGA s volatilní konfigurací ukládají konfigurační informace do paměťových buněk typu SRAM. FPGA postavená na této technologii mají jednoznačnou výhodu ve snadné konfiguraci a rekonfigurovatelnosti i za běhu systému. Další nespornou výhodou SRAM FPGA je technologický náskok; k výrobě FPGA s volatilní konfigurací je používán standardní technologický proces CMOS bez dodatečných kroků a proto jsou SRAM obvody vždy o jednu až dvě technologické generace napřed před ostatními FPGA. Není bez zajímavosti, že SRAM FPGA se pro svoji pravidelnou strukturu dnes stávají prvními návrhy vyráběnými v nových technologických procesech (například 90 nm UMC CMOS proces byl „vyladen“ na FPGA obvodech); v době psaní rukopisu byly nejnovější FPGA obvody firmy Xilinx testovány už v 28 nm procesu (řady Virtex-7, Artix, Kintex). Na druhou stranu použití SRAM paměťových buněk přináší i nevýhody. Programovatelný obvod musí být po startu systému nakonfigurován – k tomu je obvykle potřeba externí paměť, to znamená větší potřebný prostor na desce s plošnými spoji a více součástek. Systém postavený kolem SRAM FPGA také není schopen pracovat okamžitě po zapnutí napájecího napětí, konfigurace může trvat řádově do stovek milisekund. Vlastní konfigurační proces způsobuje

zvýšenou spotřebu elektrické energie po startu systému (tzv. inrush current). Na příslušný proudový odběr je pak nutné dimenzovat i napájení systému. Nevýhodou také představuje vyšší spotřeba energie za běhu zařízení; i když je FPGA v režimu s nízkým příkonem (low power mode), odebírá stále proud ze zdroje na udržení konfigurace. Konečně, u SRAM FPGA je těžší zajistit zabezpečení intelektuálního vlastnictví, protože konfiguraci obvodu lze jednoduše vyčíst z konfigurační paměti (lze ovšem použít speciální podporu pro šifrování konfigurace, tu ale nemají všechna FPGA – viz např.).

FPGA s nevolatilní konfigurací ukládá konfigurační bity v nevolatilních paměťových buňkách. Typicky se lze setkat s flash pamětí, EEPROM a s tzv. antifuses (antipojistkami). Jasná nevýhoda takového obvodu je obtížnější změna konfigurace – u antifuses je nemožná, u ostatních je s ní třeba počítat při návrhu. Výhodou je ovšem nižší spotřeba energie výsledným zařízením (chybějící konfigurační fáze, low-power mód s nižší spotřebou). Nevolatilní FPGA obvody přináší proti volatilním FPGA obvodům i vyšší odolnost proti radiaci a nejvyšší úroveň ochrany intelektuálního vlastnictví, neboť konfigurace je uložena přímo v obvodu a obvykle lze propálením příslušné propojky znemožnit i její zpětné vyčtení. I po rozpouzdření čipu na specializovaném pracovišti je téměř nemožné zjistit skutečnou konfiguraci obvodu. Jednotlivé obvody mají svoje specifika: antifuse je v nenaprogramovaném stavu rozpojená a programováním se propojují (opak pojistky) – vytvoří se rezistivní spojka. FPGA je ovšem konfigurovatelné pouze mimo cílovou aplikaci; v případě masové produkce je nutné zvážit dostupnost dostatečného množství obvodů (čas potřebný pro naprogramování obvodu může být kritický) a případně využít možnosti konverze FPGA do zákaznického integrovaného obvodu. FPGA konfigurované pomocí antifuses má výhodu ve zvýšené radiační odolnosti ve srovnání s ostatními technologiemi. Samotné antifuses ale pro zvýšení odolnosti nestačí, je vhodné je dále kombinovat s trojnásobnou redundancí registrů a majoritním dekodérem (TMR, Triple-Module Redundancy, viz např. výrobce Actel, rodina obvodů RTAX-S). Výroba FPGA obvodů s antifuses ovšem vyžaduje dodatečné kroky ke standardnímu CMOS procesu a proto jsou tyto obvody obvykle o jednu až dvě generace za SRAM FPGA, což částečně odstraňuje jejich výhodu v nižší spotřebě a vyšší rychlosti kompaktnější logické matice FPGA (antifuses jsou menší, než SRAM buňky), EEPROM/flash FPGA umožňují jak programování v aplikaci, tak před vlastním použitím. Spotřeba energie obvodem je zhruba mezi antifuse FPGA a SRAM FPGA obvody a stejně jako antifuse FPGA i EEPROM/flash FPGA obvody ke své výrobě potřebují CMOS proces s dostatečnými kroky. I tyto obvody tedy technologicky poněkud zaostávají za SRAM FPGA.

Existují i FPGA, která jsou mezi oběma protipóly – obvody s SRAM konfigurací a flash pamětí integrovanou přímo v pouzdře FPGA obvodu (viz např. Xilinx Spartan 3AN). Výhoda kombinovaného přístupu je ve zmenšení plochy a složitosti desky plošných spojů zařízení.“[1]

1.4 Návrh logických obvodů

Návrh logických obvodů do programovatelných obvodů má tyto fáze:

- Specifikace funkce obvodu
- Kódování návrhu
- Verifikace na RTL úrovni
- Implementace
- Verifikace na hradlové úrovni
- Konfigurace

Specifikace funkce obvodu je prvním a nejdůležitějším krokem při návrhu nového obvodu. Zde musí být důsledně popsány všechny funkce a chování požadovaného obvodu, které si nesmí vzájemně odporovat.

Kódování návrhu je proces, kdy přenášíme svoje požadavky na funkci a chování obvodu do počítačem srozumitelného zápisu. Zde máme více možností například formou hradlového schématu nebo pomocí RTL popisu. Hradlové schéma realizujeme v editoru schémat, toto vyjádření je velice srozumitelné a přehledné pro menší obvody, to je však snad jeho jediná výhoda. Tento návrh je časově náročný a většinou použitelný jen pro zvolený obvod. Přenos do jiného obvodu vyžaduje úpravu, což je zdlouhavé, jakož i jakákoliv jiná úprava navrženého schématu. RTL popis je dnes nejpoužívanější pro zachycení návrhu a je realizován v textovém zápisu v některém z jazyků HDL. Mezi nejpoužívanější jazyky HDL jsou VHDL a Verilog, které umožňují popis obvodu na vysoké úrovni abstrakce. Práce v jazyku HDL je velmi rychlá a přehledná, jeden řádek zdrojového kódu může v cílovém provedení obvodu reprezentovat až několik desítek hradel. Obvod popsáný v jazyku HDL lze snadno použít pro více druhů obvodů (je přenositelný), umožňuje provádět simulaci navrženého obvodu na základě téhož zdrojového textu a v případě požadavku na větší série aplikované v obvodech ASIC lze použít tento zápis pro implementaci do těchto obvodů.

Verifikaci na RTL úrovni ověřujeme prostřednictvím různých simulací správnou funkčnost navrhovaného obvodu. Ověřování obvodu je doporučeno průběžně ještě před implementací, která bývá časově náročná. K ověřování obvodů používáme funkční a časovou simulaci. Funkční simulace slouží k ověření správné funkce popisu obvodu, která je velmi rychlá ale nepočítá s časovými parametry obvodu. Časová simulace slouží k ověření časových parametrů jako je zpoždění hradel, maximální možná frekvence a dodržení potřebného předstihu a přesahu signálů.

Implementaci můžeme rozdělit do několika fází: 1. syntéza, 2. mapování a 3. rozmístění a propojení. Syntéza je proces, kdy je popis obvodu z vyšší úrovně převeden na nižší úroveň a jsou detekovány logické obvody (kombinační, sekvenční, násobičky, sčítačky) a je vytvořen netlist (seznam logických prvků a jejich vzájemného propojení). Mapování je konverze netlistu do konkrétního zvoleného obvodu FPGA, seskupování LUT tabulek, odstraňování nepoužívané

logiky. Rozmístění a propojení je proces, který nejdříve rozmístí v konfigurovatelné matici čipu jednotlivé logické prvky a poté tyto prvky pomocí PI mezi sebou propojí.

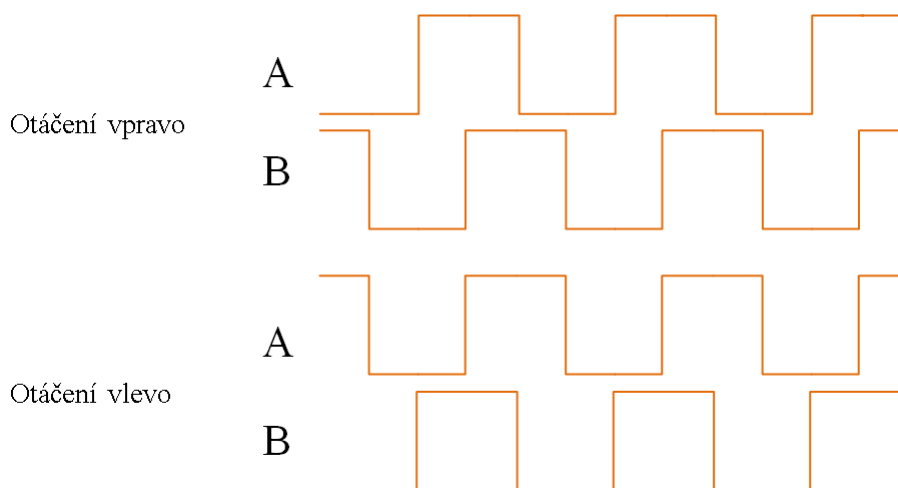
Verifikace na hradlové úrovni je nejpřesnější simulace, která dokáže odhalit chyby, které verifikace na RTL úrovni nemusí identifikovat, proto je doporučeno toto ověření spolu se statickou časovou analýzou obvodu vždy provést, i když je mnohem časově náročnější než verifikace na RTL úrovni.

Při konfiguraci je vygenerován konfigurační soubor přímo pro zvolený typ FPGA, který je pak nahrán do FPGA pomocí JTAG portu nebo po sériové lince většinou přes USB do paměti SRAM nebo EEPROM. Z těchto pamětí se pak FPGA konfiguruje na náš vytvořený obvod.

2 Přehled inkrementálních rotačních snímačů

2.1 Úvod do inkrementálních rotačních snímačů

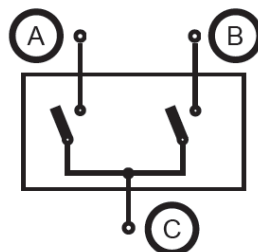
Rotační snímače jsou vlastně elektromechanickými převodníky převádějícími rotační pohyb na elektrické signály. Používají se k ovládání elektronických zařízení, např. monitorů, PC (myš), hifi soustav, měřicích přístrojů atd., ale asi nejvíce jako zpětná vazba rotačních pohonů. Výstupem kodérů jsou zpravidla dva kvadrurní signály přímo úměrné otáčení kodéru, značeny většinou A a B, které jsou mezi sebou fázově posunuty o 90° (viz obrázek č. 8). Z těchto signálů je pak možno dekódovat směr otáčení, počet kroků nebo otáček, jejich rychlost nebo třeba úhel natočení rotačního kodéru. Mezi základní parametry kodérů patří počet impulzů na otáčku, maximální rychlost otáčení a použitý výstupní obvod.



Obrázek 8: Kvadrurní výstupní signály inkrementálních rotačních kodérů

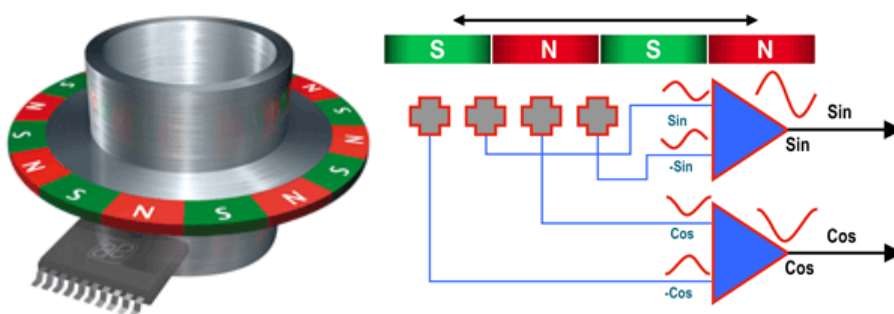
„Chyby rotačního inkrementálního senzoru se ve svém důsledku projevují na výstupních signálech- tj. signálech A a B. Některé z nich jsou dány již konstrukcí a použitou elektronikou senzoru a nelze je dodatečně odstranit. Sem patří proměnná svítivost LED a citlivost fotosenzoru, drobné nepřesnosti rozměrů a polohy průhledných a neprůhledných rysek kotouče, excentrické uložení kotouče a nepřesné umístění páru LED – fotosenzor vůči segmentům maskovacího kotouče. Další chyby jsou způsobeny okolními vlivy a nesprávným použitím inkrementálního senzoru. Zde je možné uvést překročení maximální povolené frekvence kvadrurního signálu a mechanické vibrace, zkreslením signálu vedením atd. Pokud se kotouč inkrementálního senzoru otáčí velice pomalu, popř. stojí – což je typické pro oblast velmi malých otáček a změnu směru, může být malá strmost výstupu fotosenzoru vícenásobně vyhodnocena komparátorem jako puls. Podobně se může projevit vliv vibrací působících na hřídel inkrementálního senzoru tím, že jsou generovány z hlediska skutečně vykonaného pohybu krátké nadbytečné pulsy.“[6]

Rotační snímače se dělí podle způsobu vzniku jejich signálů na mechanické, optické a magnetické. Mechanické snímače používají ke spínání mechanické kontakty, které spínají a rozpínají vývody A a B zpravidla se zemí. Jejich použití je určeno pro malé otáčky (ruční ovládání) a občasný provoz z důvodu nepříliš velké životnosti. Mezi jejich hlavní nevýhody rovněž patří generování zákmitů při spínání a rozpínání kontaktů, zato jejich cena, která začíná od desítek korun, z nich dělá jednu z nejdostupnějších součástí tohoto typu.



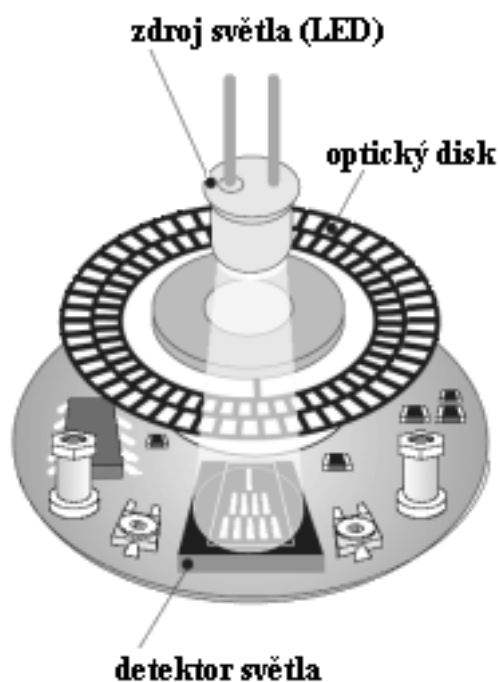
Obrázek 9: Vnitřní zapojení mechanického rotačního kodéru [16]

Magnetické snímače jsou založeny na principu Hallova jevu, kdy je na kotouči nebo kroužku magnetický materiál se střídajícím magnetickým polem, který je snímán dvěma nebo čtyřmi hallovými snímači, které podobně jako u optických snímačů převádí sinusový signál na kvadratický. Jejich výhodou je, že snímače a magnetický kotouč nebo kroužek mohou být odděleny - tzn. nejsou v jednom pouzdře. Toto lze využít při montáži na velké průměry hřídelí (například o průměru jednoho metru), kdy se na hřídel nasadí magnetický kroužek nebo pásek a na pevnou část se připevní snímač s elektronikou. Díky této vlastnosti jsou magnetické rotační kodéry velmi odolné a používají se v extrémních podmínkách.



Obrázek 100: Princip magnetického inkrementálního snímače [13]

Optické snímače jsou založeny na principu clonění světelného toku mezi zdrojem a snímačem světla, k tomu se používají skleněné nebo plastové clonící kotouče. Napěťový výstup ze snímače má při otáčení tvar sinusovky, který se dále zesílí a převede pomocí napěťových komparátorů na obdélníkový průběh. V některých aplikacích kodérů se tyto signály nepřevádějí a jsou po zesílení přivedeny přímo na výstup kodéru. Optické kodéry vynikají velkou přesností až 4.000.0000 kroků na otáčku a dlouhou životností, proto se nejčastěji používají v obráběcích strojích.

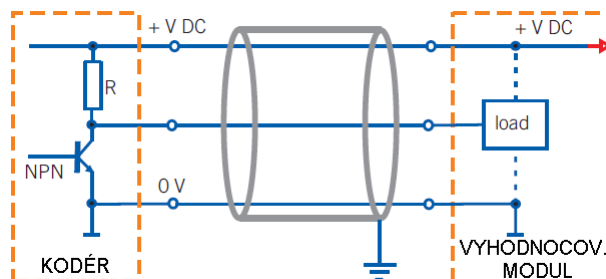


Obrázek 111: Princip optického inkrementálního kodéru [13]

2.2 Výstupní obvody inkrementálních rotačních snímačů

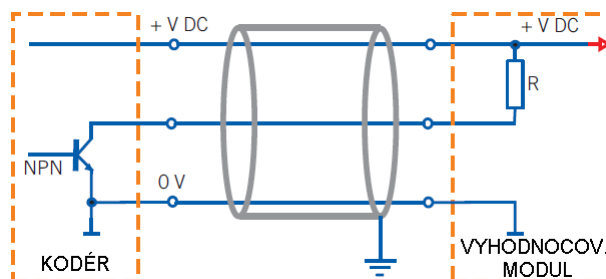
Výstupní obvody rotačních snímačů nám předurčují způsob jejich použití. Mezi nejpoužívanější výstupní obvody u optických a magnetických rotačních snímačů jsou napěťový výstup, otevřený kolektor, Push-pull a Line driver.

Napětový výstup je kompatibilní s TTL logikou a jeho přenos signálů je náchylný na délku vedení a frekvenci přenášeného signálu.



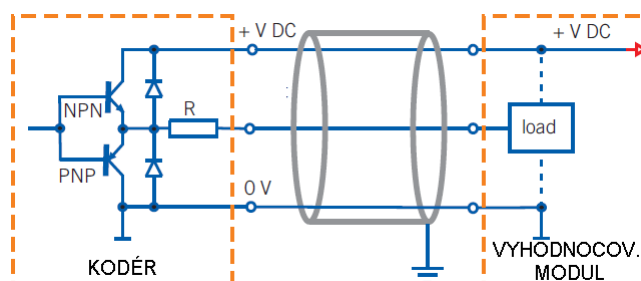
Obrázek 12: Zapojení napětového výstupu kodéru [15]

Otevřený kolektor má velmi podobné vlastnosti jako napětový výstup výhoda je jen v možnosti většího odběru proudu přes zátěž.



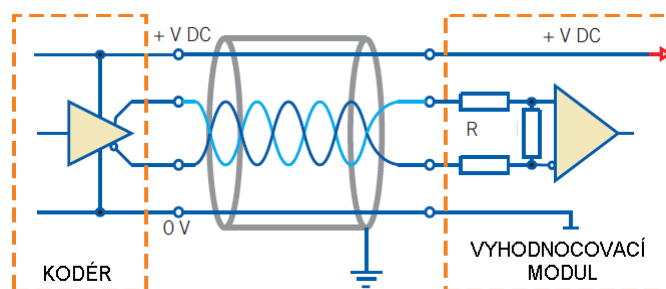
Obrázek 13: Zapojení otevřeného kolektoru kodéru [15]

Push-pull neboli komplementární zapojení tranzistorů má výhodu zvláště při větším odběru proudu, kde je možno získat na výstupu napětí téměř od nulového, až po napájecí., oproti napětového výstupu, kde se projeví úbytek na odporu R a tím se napětí snižuje. Toto zapojení zároveň zvyšuje možnost použití vyšších frekvencí pro přenos signálů.



Obrázek 14: Zapojení Push-pull kodéru [15]

Line driver používá zdvojený signál u něhož druhý je negací prvního a na konci vedení v přijímači je diferenciální zesilovač, který odfiltruje souhlasné rušivé napětí. Proto se používá zvláště v prostředí s elektromagnetickým rušením a při dlouhém vedení mezi kodérem a vyhodnocovacím modulem.



Obrázek 15: Zapojení Line driver kodéru [15]

2.3 Přehled inkrementálních rotačních snímačů

Přehled inkrementálních rotačních kodérů vychází ze situace na trhu. Vybráno bylo několik nejzajímavějších snímačů s velkým rozsahem parametrů a universálním použitím a rovněž pár specifických (např. na velké průměry hřídelí přes 700 mm, nebo navržené přímo k měření vzdálenosti pomocí kolečka, kdy je rozlišení kodéru uváděno na minimální délku měření).

Výrobce		PEWATRON	PEWATRON	PEWATRON	PEWATRON	PEWATRON	PEWATRON
Série		38	E3,E6	EC35	OMS	OEZ	SBU
Typ		38S,38H,38M					
Druh snímače		optický	optický	optický	optický	optický	optický
Rozlišení/ot	od	100	64	500	50	35	
	do	4096	2500	10000	125	1500	8192
Max. otáčky/min.	od	6000	2400	6000			
	do	10000	60000	12000	6000	6000	500
Spínací f (kHz)		240		1000	10	150	
Výst. signály		A,B,Z	A,B,Z	A,B,Z	A /A,B	A,B,Z./A./B./Z	A,B,Z./A./B./Z, F0,F1,F2,F3,F4
Druh výst. sign.		kvadraturní	kvadraturní	kvadraturní	sin,cos	kvadraturní	kvadraturní
Typ hřídele		plná/ dutá	kotouče	dutá	plná	plná	
Ø hřídele	od	5	2	5			
	do	8	25	10	1.5	4.5	50
Pracovní teplota (°C)	od		-40	-20	0	-10	-10
	do		100	105	50	60	70
Krití		IP 50			IP 50	IP 50	IP 50
Odběr sním. (mA)		30 - 60	17 - 85	13 - 44			
Druh výst. obvodu		Napájecí napětí (V)					
Line driver (RS422)		4.5 - 13.2	5			4.75 - 5.25	11.4 - 12.6
Otevřený kolekt.		4.5 - 30		30	20	4.5 - 26.2	
Napět'ový výstup						4.5 - 26.2	
Push-pull		4.5 - 30					
		Maximální zatížení výstupu (mA)					
Line driver (RS422)		20				20	20
Otevřený kolekt.		40			20	20	
Napět'ový výstup						20	
Push-pull		600					

Tabulka 1: Přehled rotačních kodérů [17]

Výrobce		ELTRA	ELTRA	ELTRA	ELTRA	ELTRA	ELTRA
Série		EL58	EL53	EH150	EMI22	EX80	EV
Typ		B, C, H, T	A, B	P	A	A, D	A, B
Druh snímače		optický	optický	optický	magn.	optický	optický
Rozlišení/ot	od	1	1	1024	2	1	1
	do	10000	10000	5040	2048	10000	10000
Max. otáčky/min.	od	3000	3000				
	do	6000	6000	1000	10000	3000	
Spínací f (kHz)		300	100	100	200	100	100
Výst. signály		A,B,Z	A,B,Z	A,B,Z	A,B,Z	A,B,Z	A,B
Druh výst. sign.		kvadraturní	kvadraturní	kvadraturní	kvadraturní	kvadraturní	kvadraturní
Typ hřídele		plná	dutá	dutá	dutá	plná	plná
Ø hřídele	od	6	6	50	6	10	6
	do	12	10	78	10	10	10
Pracovní teplota (°C)	od	0	0	0	-25	-20	0
	do	60	60	60	125	50	60
Krití		IP 54/ IP 66	IP 54/ IP 66	IP 54/ IP 65	IP 68	IP 65	IP 64
Odběr sním. (mA)		100	100	100	100	100	100
Druh výst. obvodu		Napájecí napětí (V)					
Line driver (RS422)		5-28	5-28	5/8-24	5	5-28	5-28
Otevřený kolekt.		5-28	5-28	5/8-24		5-28	5-28
Napět'ový výstup		5-28	5-28	5/8-24	5	5-28	5-28
Push-pull		5-28	5-28	5/8-24		5-28	5-28
		Maximální zatížení výstupu (mA)					
Line driver (RS422)		20	20	20	15	20	20
Otevřený kolekt.		50	50	50		50	50
Napět'ový výstup		50	50	50	15	50	50
Push-pull		50	50	50		50	50

Tabulka 2: Přehled rotačních kodérů [15]

Výrobce		ELTRA	ELTRA	ENCODER	ENCODER	ENCODER	Autonics
Série		ETM	ES				E50S
Typ		R	K	25T	770	771	
Druh snímače		magn.	optický	optický	optický	optický	optický
Rozlišení/ot	od	64		1	1	1	15
	do	4096	2048	10000	4096	4096	8000
Max. otáčky/min.	od			4000		3500	
	do	8000	6000	8000	6000	6000	5000
Spínací f (kHz)		350	205	1000	200	200	180
Výst. signály		A,B,Z	A,B,C,D,Z	A./A,B./B,Z./Z	A./A,B./B,Z./Z	A./A,B./B,Z./Z	A./A,B./B,Z./Z
Druh výst. sign.		kvadrurní	sin,cos	kvadrurní	kvadrurní	kvadrurní	kvadrurní
Typ hřídele		kotouč	dutá	dutá	dutá	dutá	plná
Ø hřídele	od	9		6	14	28	
	do	35	12.7	28	24	43	8
Pracovní teplota (°C)	od	-20	-20	-20	0	0	-10
	do	85	110	105	100	100	70
Kriti		IP 67	IP 64	IP 66	IP 65	IP 65	IP 50
Odběr sním. (mA)		40	130	100	100	100	60
Druh výst. obvodu		Napájecí napětí (V)					
Line driver (RS422)		5/8-24		5-28	5-28	5-28	5/12-24
Otevřený kolekt.			5	5-28	5-28	5-28	5/12-24
Napětový výstup				5-28	5-28	5-28	
Push-pull		5/8-24		5-28	5-28	5-28	
Druh výst. obvodu		Maximální zatížení výstupu (mA)					
Line driver (RS422)		20		20	20	20	20
Otevřený kolekt.			20	20	100	100	30
Napětový výstup				20	100	100	
Push-pull		20		20	20	20	

Tabulka 3: Přehled rotačních kodérů [15], [18], [19]

Výrobce		Autonics	Omron	Omron	Omron	Automation direct	Automation direct
Série		ENC	E6A2	E6B2	E6C2	TRD	TRD
Typ						SXXXXX	GXXXXX
Druh snímače		optický	optický	optický	optický	optický	optický
Rozlišení/ot	od	1 mm	10	10	10	100	100
	do	1 m	360	2000	2000	2500	1000
Max. otáčky/min.	od						
	do	5000	5000	6000	6000	6000	5000
Spínací f (kHz)		180	30	100	100	200	100
Výst. signály		A,B	A./A,B./A,B,Z	A,B,Z	A./A,B./B,Z./Z	A./A,B./B,Z./Z	A,B,Z
Druh výst. sign.		kvadrurní	kvadrurní	kvadrurní	kvadrurní	kvadrurní	kvadrurní
Typ hřídele		kolečko	plná	plná	plná	plná	plná
Ø hřídele	od			6	6		
	do	kolečko	4	10	8	8	10
Pracovní teplota (°C)	od	-10	-10	-10	-10	-10	-10
	do	70	55	70	70	70	70
Kriti		IP 50	IP 50	IP 50	IP 64	IP 65	IP 65
Odběr sním. (mA)		60	50	100	160	50	
Druh výst. obvodu		Napájecí napětí (V)					
Line driver (RS422)				5	5	5	
Otevřený kolekt.		5/12-24	5-24	5-24	5-24	12-24	
Napětový výstup			5-12	5-12	5-12		10-30
Push-pull							
Druh výst. obvodu		Maximální zatížení výstupu (mA)					
Line driver (RS422)							
Otevřený kolekt.				20	30		
Napětový výstup		30	30	35	35		
Push-pull			20	20	20		30

Tabulka 4: Přehled rotačních kodérů [20], [21]

Výrobce	LENORD+ BAUER	LENORD+ BAUER	Agilent Technologies	Agilent Technologies	SIKO	SIKO
Série			HEDS	HEDS		
Typ	GEL 293	GEL 2010	5XXX	9040/9140	IG04M	IG07
Druh snímače	magn.	magn.	optický	optický	magn.	optický
Rozlišení/ot	od		2	50	96	2
	do	10000	1024	1024	1024	2000
Max. otáčky/min.	od				600	
	do	8000	6000	30000	30000	3000
Spínací f (kHz)	200	200	100	100	100	25
Výst. signály	A _z /A _z B _z /B _z Z _z /Z _z	A _z /A _z B _z /B _z Z _z /Z _z	A _z B _z Z _z	A _z B _z Z _z	A _z /A _z B _z /B _z Z _z /Z _z	A _z B _z Z _z
Druh výst. sign.	kvadraturní	kvadraturní	kvadraturní	kvadraturní	kvadraturní	kvadraturní
Typ hřídele	dutá	plná	dutá	-	dutá	dutá
Ø hřídele	od		2	2	10	
	do	16	10	8	14	20
Pracovní teplota (°C)	od	-20	-20	-40	-40	-20
	do	85	70	100	100	80
Krití		IP 66	IP 67		IP 63	IP 54
Odběr sním. (mA)				57		
Druh výst. obvodu	Napájecí napětí (V)					
Line driver (RS422)		5				5/10-30
Otevřený kolekt.		10-30		5		10-30
Napětíový výstup			5			10-30
Push-pull						
Maximální zatížení výstupu (mA)						
Line driver (RS422)						40
Otevřený kolekt.						80
Napětíový výstup			40			40
Push-pull						

Tabulka 5: Přehled rotačních kodérů [22], [23], [24]

Výrobce	PEPPERL+ FUCHS	PEPPERL+ FUCHS	PEPPERL+ FUCHS	PEPPERL+ FUCHS	PEPPERL+ FUCHS	Kübler
Série	TVI40N	RVI58N	RHI90N	RVI84N	MNI40N	37X0
Typ						
Druh snímače	optický	optický	optický	optický	mag.	optický
Rozlišení/ot	od	50	100	100	1	50
	do	1024	10000	2500	25	3200
Max. otáčky/min.	od					
	do	6000	12000	3500	3000	30000
Spínací f (kHz)	100	200	120	5	1000	250
Výst. signály	A _z /A _z B _z /B _z Z _z /Z _z	A _z /A _z B _z /B _z Z _z /Z _z	A _z /A _z B _z /B _z Z _z /Z _z	A _z B _z	A _z /A _z B _z /B _z Z _z /Z _z	A _z /A _z B _z /B _z Z _z /Z _z
Druh výst. sign.	kvadraturní	kvadraturní	kvadraturní	kvadraturní	kvadraturní	kvadraturní
Typ hřídele	plná	plná	dutá	plná	kotouč	plná/dutá
Ø hřídele	od	6	6	16		6
	do	8	12	45	10	15
Pracovní teplota (°C)	od	-10	-20	-20	-20	-25
	do	70	80	70	60	100
Krití		IP 54	IP 65	IP 65	IP 65	IP 67
Odběr sním. (mA)		55	150	70		
Druh výst. obvodu	Napájecí napětí (V)					
Line driver (RS422)	5	5/10-30		5		5
Otevřený kolekt.						
Napětíový výstup						
Push-pull	4.75-30	10-30	10-30		10-30	5-30
Namur				8		
Maximální zatížení výstupu (mA)						
Line driver (RS422)	30	20	20			20
Otevřený kolekt.						
Napětíový výstup						
Push-pull	30	40	40			20
Namur				3		

Tabulka 6: Přehled rotačních kodérů [25], [26]

Výrobce	Kübler		LEINE LINDE	LEINE LINDE	LEINE LINDE	LEINE LINDE	GIVI MISURE
Série							
Typ	RI20/LI20		ESI 1503	Atex 632	865	RSD 525 63	EN500,EN600
Druh snímače	mag.		optický	optický	optický	optický	optický
Rozlišení/ot	od		1		150	1	8
	do	3600	10000	10000	6350	10000	25000
Max. otáčky/min.	od						
	do	12000	4000	6000	4000	6000	8000
Spínací f (kHz)	250		300	100	200/100	300	300
Výst. signály	A _z /A _z B _z /B _z Z _z /Z		A _z B _z Z	A _z /A _z B _z /B _z Z _z /Z	A _z /A _z B _z /B _z Z _z /Z	A _z /A _z B _z /B _z Z _z /Z	A _z /A _z B _z /B _z Z _z /Z
Druh výst. sign.	kvadrurní		kvadrurní	kvadrurní	kvadrurní	kvadrurní	kvadrurní
Typ hřídele	dutá		plná	plná	dutá	plná	plná
Ø hřídele	od		11	6	12		6
	do	30	15	10	16	10	10
Pracovní teplota (°C)	od	-20	-40	-20	-25	-40	0
	do	80	70	60	70	70	70
Krití	IP 65		IP 67	IP 65	IP 65	IP 67	IP 65
Odběr sním. (mA)			75		85		110
Druh výst. obvodu	Napájecí napětí (V)						
Line driver (RS422)	4.8-26		5/9-30	4-8	5	5/9-30	24
Otevřený kolekt.							5-12
Napět'ový výstup			5/9-30			9-30	5-12
Push-pull	4.8-26		5/9-30		9-30		12-24
	Maximální zatížení výstupu (mA)						
Line driver (RS422)			20	20	20		30
Otevřený kolekt.							30
Napět'ový výstup			40				40
Push-pull			40		40		30

Tabulka 7: Přehled rotačních kodérů [26], [27], [28]

Výrobce	Megatron		Megatron	Megatron	Megatron	Megatron	Baumer
Série	M101B		MHL40	MHU60	MRXC50	SPDH	
Typ							AG14
Druh snímače	optický		optický	optický	optický	optický	optický
Rozlišení/ot	od	25	10	25		100	250
	do	128	3600	100	50	360	2500
Max. otáčky/min.	od					10000	
	do		5000	200	120	36000	
Spínací f (kHz)	0.5		180	5	0.1	30	120
Výst. signály	A,B		A _z /A _z B _z /B _z Z _z /Z	A _z /A _z B _z /B _z Z _z /Z	A _z B	A _z B	A _z /A _z B _z /B _z Z _z /Z
Druh výst. sign.	kvadrurní		kvadrurní	kvadrurní	kvadrurní	kvadrurní	kvadrurní
Typ hřídele	plná		dutá	ovl. kolečko	plná	dutá	dutá
Ø hřídele	od		6			1.5	
	do	6	12		6	6	70
Pracovní teplota (°C)	od	0	-10	-10	0	-20	-20
	do	60	70	60	60	85	85
Krití	IP 40		IP 50	IP 40	IP 40	IP 40	IP 54
Odběr sním. (mA)			60	150	20	29	100
Druh výst. obvodu	Napájecí napětí (V)						
Line driver (RS422)			5				5
Otevřený kolekt.			5/11.4-25.2				
Napět'ový výstup	5		5/11.4-25.2	5-12	5	5	
Push-pull			5/11.4-25.2				9-26
	Maximální zatížení výstupu (mA)						
Line driver (RS422)				20			
Otevřený kolekt.							
Napět'ový výstup				20			
Push-pull							

Tabulka 8: Přehled rotačních kodérů [29], [30]

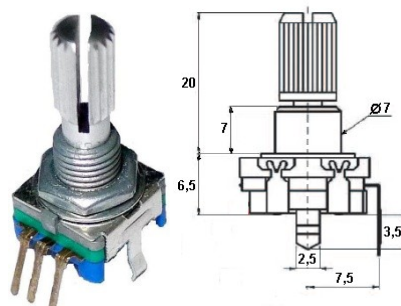
Výrobce		Baumer	Baumer	Baumer	Baumer	Baumer	Baumer
Série							
Typ		BDH,BDT	BDT sine	BHF,BHG	BHK	BRIV 58K,58S	EEx HOG 161
Druh snímače		optický	optický	optický	Optický	mag.	optický
Rozlišení/ot	od	7200	1000	4096	360	64	250
	do	320000	5000	320000	5120	2048	2500
Max. otáčky/min.	od	6000	6000	3000		6000	
	do	12000	12000	6000	12000	12000	5600
Spínací f (kHz)		1300	160	1300		320	120
Výst. signály		A _z /A _z B _z /B _z Z _z /Z	A _z B _z Z	A _z /A _z B _z /B _z Z _z /Z	A _z /A _z B _z /B _z Z _z /Z	A _z /A _z B _z /B _z Z _z /Z	A _z /A _z B _z /B _z Z _z /Z
Druh výst. sign.		kvadrturní	sin,cos	kvadrturní	kvadrturní	kvadrturní	kvadrturní
Typ hřídele		plná	plná	dutá	Dutá	plná	dutá
Ø hřídele	od	6				6	30
	do	10	6	12	6	10	70
Pracovní teplota (°C)	od	-20	-20	-20	-20	-20	-20
	do	85	85	85	85	85	70
Krití		IP 42/ IP 65	IP 42/ IP 65	IP 42/ IP 65	IP 64	IP 42/ IP 65	IP 56
Odběr sním. (mA)		80	100	60	60	30	100
Druh výst. obvodu		Napájecí napětí (V)					
Line driver (RS422)		5		5	5		5/9-30
Otevřený kolekt.							
Napětový výstup			5			5	
Push-pull		10-30		10-30	4.5-30	10-30	
		Maximální zatížení výstupu (mA)					
Line driver (RS422)		20					
Otevřený kolekt.							
Napětový výstup						20	
Push-pull		30				30	

Tabulka 9: Přehled rotačních kodérů [30]

Výrobce		Baumer		Baumer		Baumer		Baumer		Baumer		SCANCON	
Série												2REB	
Typ		HG18,HG22		HOG 86 L		POG 10 + DSL		MHGP 400		MHGE 800			
Druh snímače		optický		optický		optický		mag.		mag.		optický	
Rozlišení/ot	od	250		500		512		256		512		4	
	do	4000		2500		2500		524288		32768		7500	
Max. otáčky/min.	od					2900							
	do	12000		10000		6000		2000		1000		9000	
Spínací f (kHz)		120		120		120		2000		300		200	
Výst. signály		A,/A,B,/B,Z,/Z		A,B,Z		A,/A,B,/B,Z,/Z		A,/A,B,/B,Z,/Z		A,/A,B,/B,Z,/Z		A,/A,B,/B,Z,/Z	
Druh výst. sign.								kvadraturní/sin .cos		kvadraturní/sin .cos		kvadraturní	
Typ hřídele		dutá		dutá		plná		Kotouč		kotouč		plná	
Ø hřídele	od	65		12				70		650			
	do	120		17		11		340		740		6	
Pracovní teplota (°C)	od	-30		-20		-30		-20		-40		-10	
	do	70		70		85		85		100		70	
Kriti		IP 54		IP 66		IP 66		IP 68		IP 68		IP 65	
Odběr sním. (mA)		100		300				160		70		35	
Druh výst. obvodu		Napájecí napětí (V)											
Optický				9-30									
RS485						15-30							
Line driver (RS422)		5/9-26						4.5-30		4.75-30			
Push-pull												5-24	
		Maximální zatížení výstupu (mA)											
Line driver (RS422)													
Push-pull												25	

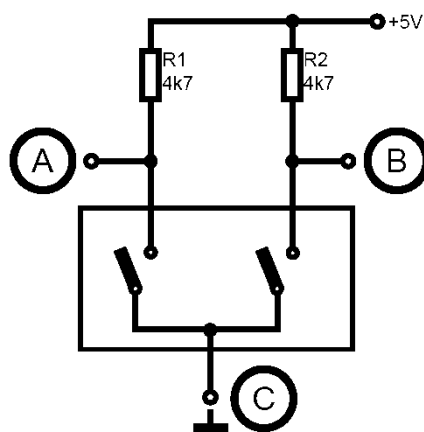
Tabulka 10: Přehled rotačních kodérů [19], [30]

2.4 Rotační kodér P-RE20



Obrázek 16: Inkrementální rotační kodér P-RE20 [16]

P-RE20 je mechanický rotační kodér, jehož velmi podobný typ se používá i na některých deskách FPGA. Rotační kodér má 20 kroků na otáčku s maximální délkou zákmitů kontaktů 5 ms. Největší dovolené proudové zatížení kontaktů je maximálně 10 mA a při jeho překračování se výrazně snižuje životnost kodéru. Je určen do prostředí s teplotou od -30°C do 70°C s relativní vlhkostí od 25 % do 85 %. Jeho životnost se udává v počtu otočení, která je u toho typu 15000 což není mnoho, proto je určen k občasnému provozu. V aretované poloze jsou kontakty uvnitř kodéru rozpojené a výstupní signály podle typického zapojení mají hodnotu logické jedničky.



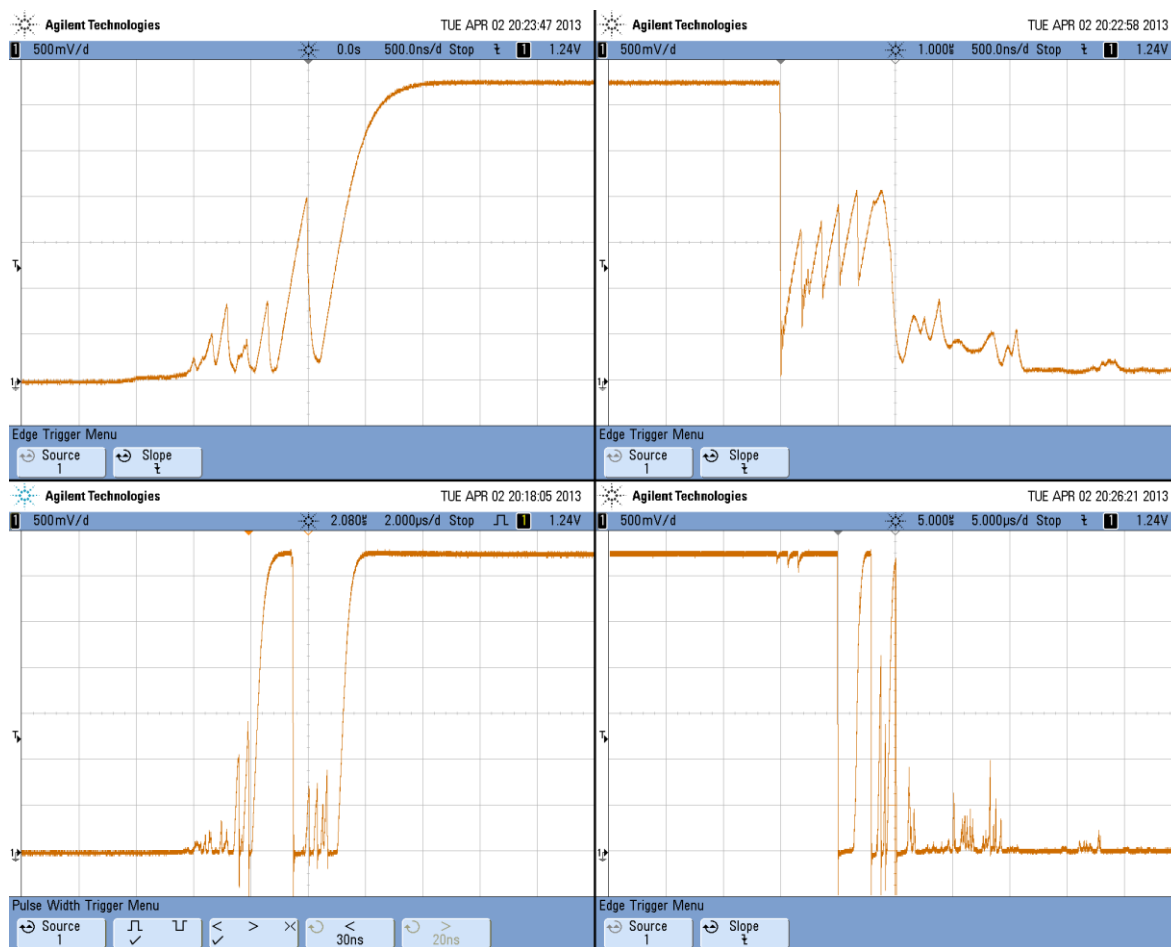
Obrázek 17: Typické zapojení kodéru P-RE20 [16]

3 Návrh číslicové jednotky pro čtení rotačního kodéru

Dekódování kvadraturních signálů z rotačního kodéru je možno několika způsoby od postavení svého nebo použití již hotových integrovaných obvodů přes mikroprocesory až k obvodům FPGA, kdy každé řešení má své výhody, nevýhody a svá omezení. Vlastní obvod složený z integrovaných obvodů je použitelný a efektivní snad jen pro zjištění informace o směru otáčení, maximálně s připojeným čítačem i informace o stavu pootočení kodéru. K tomuto účelu je výhodnější použít již hotový, cenově dostupný integrovaný obvod například LS7083, nebo HCTL-2032, který dokáže s frekvencí 33 MHz dekodovat paralelně signály ze dvou kodérů a má v těle integrovaný 32-bitový obousměrný čítač. Toto řešení je vhodné pro malé aplikace, kdy nepožadujeme měnit parametry (jako například frekvenci snímání, počet stejných vzorků pro platný signál, parametry čítače) nebo řešit další složitější operace s výsledky dekodování. Některé obvody umožňují částečné nastavení - například LS7083 u kterého lze nastavit minimální délku pulsu z kodéru (od 400 ns do 25 μ s), který bude vyhodnocen jako platný. Pro složitější obvody kdy je třeba měnit parametry, dělat složitější výpočty např. pro řízení pohonů a ovládání jiných zařízení závislé na dekodování kodérů je vhodné použít mikroprocesory, které tyto složitější operace bez obtíží zvládají. Omezení je však ve snímání maximální frekvence kvadraturního signálu, která je závislá na taktovací frekvenci mikroprocesorů, jenž se pohybuje v jednotkách až stovkách kHz, a hlavně v sériovém zpracování vstupních signálů. Proto výběr pro návrh číslicové jednotky padl do programovatelných polí, které mají taktovací frekvenci v jednotkách až stovkách MHz a hlavně umožňují paralelní zpracování signálů, které z nich dělá ideální zařízení pro zpracování signálů z inkrementálních rotačních kodérů.

3.1 Měření

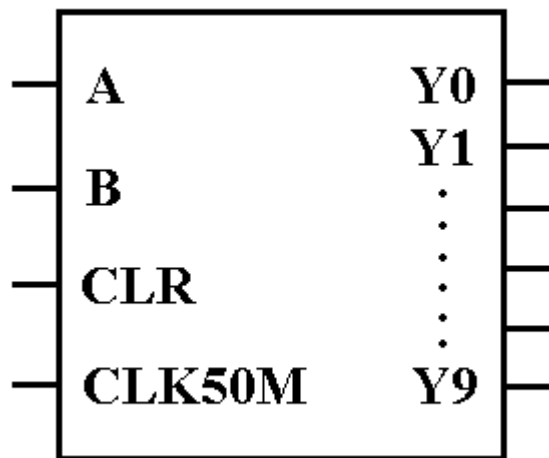
Základem pro kvalitní návrh obvodu, je získat co nejvíce informací o vstupních signálech určených pro zpracování a vyhodnocování, proto prvním krokem bylo zjistit průběh výstupních signálů z rotačního kodéru P-RE20. Pomocí osciloskopu byly změřeny zákmity vznikající při spínání a rozpínání mechanických kontaktů kodéru, které jsou vidět na obrázku č. 18. Z měření vyplývá, že zákmity, které by ovlivnily určení správné logické hodnoty na vstupu FPGA, se pohybují v délce od 0,5 μ s do 100 μ s. Toto je podstatně méně, než je uvedeno v datovém listu kodéru, ve kterém se udává do 5 ms. Nejdelší jednotlivé impulzy v zákmitěch nepřesahovaly 15 μ s a střída kvadraturních signálů je 1:1. Měřením bylo také zjištěno, že při vyšších otáčkách se délka zákmitů snižuje.



Obrázek 18: Výstupní signály se zákmity při spínání a rozpínání kontaktů kodéru P-RE20

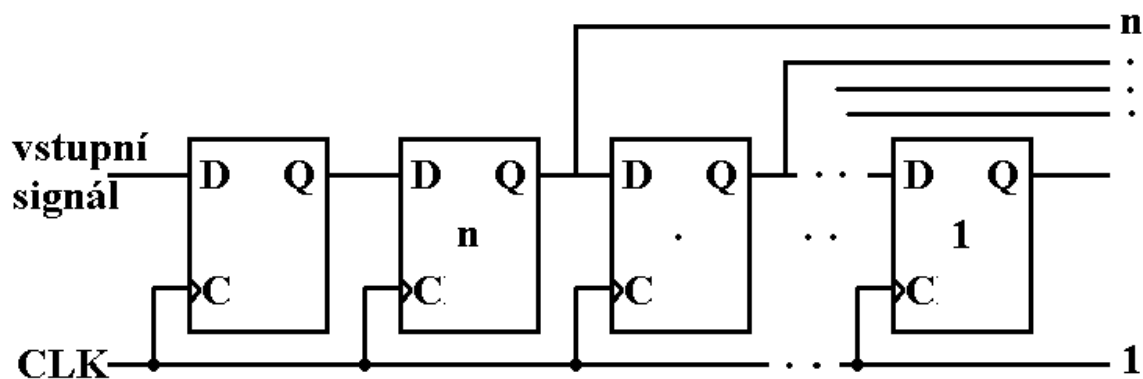
3.2 Návrh

Navrhovaný obvod bude v závislosti na vstupních signálech z rotačního kodéru určovat počet jeho kroků od 0 do 999. Při otáčení doprava se kroky budou přičítat, při otáčení doleva odečítat, při přetečení čítače se jeho stav vynuluje a při podtečení se nastaví na hodnotu 999. V FPGA bude navrhovaný obvod vypadat podle obrázku č. 19, vstupem budou signály A a B z rotačního kodéru, hodinový signál o frekvenci 50 MHz a asynchronní reset CLR, který umožní nastavit obvod do výchozího stavu a vynulování čítače kroků a zároveň bude vyveden do fyzického tlačítka na desce FPGA. Výstupem pak bude binární hodnota obousměrného čítače kroků rotačního kodéru. Vytvořený obvod bude muset umět zesynchronizovat vstupní signály, poté je vyfiltrovat a nakonec správně dekodovat.



Obrázek 19: Blokový symbol představující navržený obvod

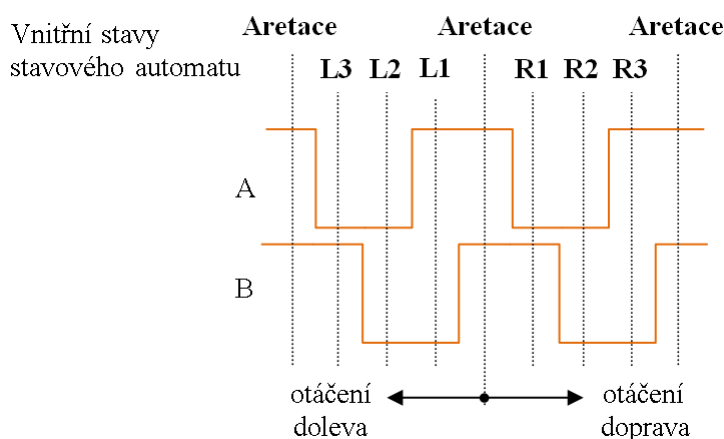
Synchronizace vstupních signálů je důležitá, protože signály A a B jsou asynchronní a při dalším zpracování těchto signálů sekvenčními obvody v FPGA by mohlo dojít při jejich vyhodnocování ke změně logické úrovně těchto vstupních signálů a tudíž k nepředvídatelnému výsledku sekvenčního obvodu. Proto musíme zajistit stabilní logickou úroveň při zpracování sekvenčních obvodů, a to synchronizací těchto vstupních asynchronních signálů A a B pomocí dvou vzorkovacích obvodů. Každý vzorkovací obvod bude složen ze dvou za sebou zapojených klopných obvodů typu D, které zapisují na výstup hodnotu jejich vstupu při náběžné hraně nastavitelného řídicího signálu CLK.



Obrázek 20: Vzorkování a filtrace pomocí posuvného registru

Filtrace zesynchronizovaného signálu bude tvořena posuvným registrem s nastavitelným počtem klopných obvodů typu D, který určuje minimální počet po sobě navzorkovaných stavů se stejnou logickou hodnotou pro určení platného vstupního stavu. V posuvném registru se porovnávají jednotlivé výstupy klopných obvodů a až při všech stejných logických úrovních je na výstup přivedena jejich logická hodnota. Filtrací se zajistí odstranění krátkých zákmitů, které by mohly při následném dekódování být vyhodnoceny jako otáčení rotačního kodéru. Vhodným zvolením vzorkovací frekvence a počtem klopných obvodů typu D ve filtru by se mělo docílit signálů, které by měly vypadat téměř jako na obrázku č. 8.

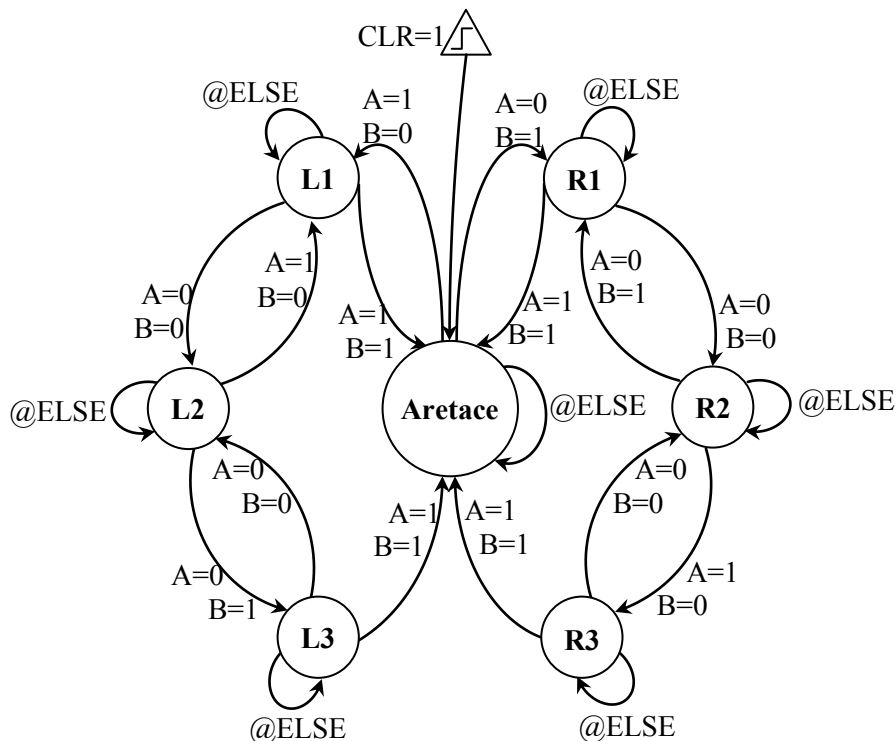
Dekódování signálů bude vycházet ze všech možných kombinací signálů A a B, které při otáčení kodéru na obě strany vznikají. Výchozí stav je Aretace a ostatní stavy jsou pojmenovány podle směru otáčení. Při otáčení doprava začínají písmenem R podle anglického slova „right“ a při otáčení doleva začínají písmenem L podle „left“, jak je vidět na obrázku č. 21.



Obrázek 21: Stavy automatu při různém směru otáčení

Samotné dekódování se bude řešit stavovým automatem Mealyho typu - jeho výstup je závislý nejen na jeho vnitřním stavu, ale i na stavu vstupních signálů. Vstupy do stavového automatu tvoří vyfiltrované signály A a B z rotačního kodéru a výstupem bude povel pro ovládání čítače kroků. Vnitřních stavů bude celkem sedm, jeden výchozí (Aretace) a tři stavy na každý směr otáčení, doprava (R1, R2, R3) a doleva (L1, L2, L3). V závislosti na kombinaci vstupních signálů z rotačního kodéru bude určena poloha i směr otáčení. Při zapnutí obvodu nebo při asynchronním resetu se automat nastaví do výchozího stavu Aretace. Pokud se bude rotační kodér otáčet doprava, změní se stav signálu A z logické 1 na 0 a signál B zůstává v logické 1, tím se automat dostane do stavu R1. Ze stavu R1 se stálým pootáčením doprava změní signál B z logické 1 do 0 a signál A zůstává v logické 0, tím se automat dostane do stavu R2, nebo při zpětném pootočení doleva se změní signál A zpátky z logické 0 na 1 a tím se automat dostane do výchozího stavu Aretace. Ze

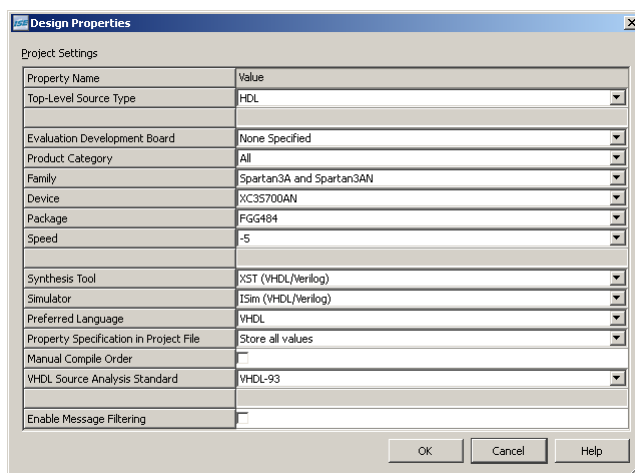
stavu R2 se stálým pootáčením doprava změní signál A z logické 0 do 1 a signál B zůstává v logické 0, tím se automat dostane do stavu R3, nebo při zpětném pootočení doleva se změní signál B zpátky z logické 0 na 1 a tím se automat dostane do stavu R1. Ze stavu R3 se stálým pootáčením doprava změní signál B z logické 0 do 1 a signál A zůstává v logické 1, tím se automat dostane opět do výchozího stavu Aretace a současně se inkrementuje čítač kroků, nebo při zpětném pootočení doleva se změní signál A zpátky z logické 1 na 0 a tím se automat dostane do stavu R2. Při otáčení doleva ze stavu Aretace se mění signál B z logické 1 na 0 a signál A zůstává v logické 1, tím se automat dostane do stavu L1. Ze stavu L1 se stálým pootáčením doleva změní signál A z logické 1 do 0 a signál B zůstává v logické 0, tím se automat dostane do stavu L2, nebo při zpětném pootočení doprava se změní signál B zpátky z logické 0 na 1 a tím se automat dostane do výchozího stavu Aretace. Ze stavu L2 se stálým pootáčením doleva změní signál B z logické 0 do 1 a signál A zůstává v logické 0, tím se automat dostane do stavu L3, nebo při zpětném pootočení doprava se změní signál A zpátky z logické 0 na 1 a tím se automat dostane do stavu L1. Ze stavu L3 se stálým pootáčením doleva změní signál A z logické 0 do 1 a signál B zůstává v logické 1, tím se automat dostane opět do výchozího stavu Aretace a současně se dekrementuje čítač kroků, nebo při zpětném pootočení doprava se změní signál B zpátky z logické 1 na 0 a tím se automat dostane do stavu L2. Z toho je patrné, že přechod mezi jednotlivými stavy automatu je dán jak aktuální hodnotou logických stavů signálů A a B, tak i aktuálním stavem automatu. Při jiném stavu než je popsáno, například při současné změně signálů A i B, ke kterému by teoreticky nemělo dojít, automat zůstane v aktuálním stavu, a čeká na dvě možné kombinace, při kterých se může dostat do jiného stavu.



Obrázek 22: Stavový automat Mealyho typu

4 Implementace návrhu do Xilinx FPGA

Pro implementaci návrhu do FPGA od firmy Xilinx byl použit software ISE Design Suite verze 13.1 od téže firmy. Prvním krokem bylo vytvoření a nastavení vlastností projektu, jako je typ FPGA. V tomto případě byl použit Spartan 3AN XC3S700AN, preferovaný jazyk VHDL v revizi z roku 1993.



Obrázek 23: Tabulka vlastnosti návrhu

Dále byl vytvořen nový zdrojový dokument typu VHDL module, ve kterém se již pomocí jazyku VHDL definoval samotný obvod. V entitě se nadefinovaly vstupy a výstupy obvodu a jejich datový typ a v architektuře vnitřní signály, konstanty a proměnné.

4.1 Vzorkovací obvod s filtrem

První část definovaného obvodu tvoří vzorkování vstupních signálů spolu s číslicovým filtrem, za použití posuvného registru. K definování obvodu byl použit proces, ve kterém se příkazy na rozdíl od jiných příkazů uvedených mimo něj vykonávají sekvenčně, i když navenek se proces chová jako paralelní příkaz. Proces je uveden ve výpisu č. 1 a má v závorce (tj. citlivostním seznamu) uveden signál CLK. To znamená, že při jakékoliv změně tohoto signálu je spuštěn tento proces. V podmínce IF jsou pak podmíněny příkazy, které se vykonají vždy při náběžné hraně signálu CLK. V prvních dvou řádcích podmínky jsou dva posuvné registry o velikosti $n+1$ kde „ n “ je nastavitelný parametr, který určuje počet stejných navzorkovaných po sobě jdoucích hodnot pro určení platné hodnoty vstupního signálu. Zde je také vidět vložení nově navzorkované logické hodnoty signálů A a B a posunutí prvních n hodnot zleva o jedno místo doprava. Na dalších řádcích je popsán filtr, kdy se porovnává n nejstarších hodnot zprava s hodnotami signálů High a Low, které reprezentují hodnoty n logických jedniček a n logických nul. Při jejich rovnosti se pak zapíše platně navzorkovaná hodnota signálu na své místo do společného signálu ABfiltr, který reprezentuje platně navzorkované a vyfiltrované hodnoty signálů A a B.


```

Filtr: process (CLK)
begin
    if CLK'event and CLK = '1' then
        Afiltr <= A & Afiltr (0 to n-1);
        Bfiltr <= B & Bfiltr (0 to n-1);
        if Afiltr(1 to n) = High then ABfiltr(1) <= '1';
        elsif Afiltr(1 to n) = Low then ABfiltr(1) <= '0';
        end if;
        if Bfiltr(1 to n) = High then ABfiltr(0) <= '1';
        elsif Bfiltr(1 to n) = Low then ABfiltr(0) <= '0';
        end if;
    end if;
end process;

```

Výpis 1: Popis obvodu vzorkování a filtrace v jazyce VHDL

4.2 Stavový automat

Ve výpisu č. 2 je definovaný stavový automat v jazyku VHDL, který je již popsán v kapitole 3.2. Za zmínku stojí výpis ze syntézy, při kterém byl rozpoznán stavový automat (anglicky Finite State Machine) a ve výpisu č. 3 je vidět výpis a způsob zakódování jeho stavů. Syntéza navrhla kódování one-hot, kdy je každému stavu přiřazena hodnota s jednou logickou jedničkou v binárním řetězci o délce odpovídající počtu stavů automatu. Toto kódování má výhodu v nízké Hemmingově vzdálenosti mezi všemi stavy, které je rovno dvěma. Toto číslo udává počet rozdílných logických hodnot bitů mezi dvěma zakódovanými stavy a zároveň určuje nutný počet klopných obvodů, které se musí překlopit při přechodu z jednoho stavu do druhého. Čím méně je překlopení, tím je menší spotřeba obvodu stavového automatu. Nevýhodou je velká délka zakódovaných stavů a tím i velký počet klopných obvodů potřebných pro stavový automat. Kódování one-hot je vhodné pro rozsáhlé stavové automaty s velkým větvením.

Existují i jiné způsoby kódování stavových automatu jako například 2-hot kódování, Binární kódování, Grayovo kódování, Johnsonovo kódování aj. 2-hot kódování je obdoba one-hot kódování, ale v každém názvu stavu jsou dvě logické jedničky, takže je téměř mezi všemi stavy zachována Hemmingova vzdálenost rovná dvěma, ale snížila se délka názvu stavů a tím i počet klopných obvodů nutných pro sestavení stavového automatu. Binární kódování se vyznačuje tím, že každý stav automatu je označen číslem v binárním tvaru, což má za výhodu rychlejší provoz a menší počet klopných obvodů pro stavový automat. Nevýhodou tohoto kódování je, že mezi některými stavy je velká Hemmingova vzdálenost. Grayovo kódování má stejnou délku názvu stavů jako binární kódování, ale vyznačuje se tím, že oba sousední stavy mají Hemmingovou vzdálenost rovnou jedné a tím je vhodný pro kruhové automaty. Johnsonovo kódování je obdoba Grayova, liší se pouze v jiném pořadí hodnot stavů, které se kóduje tak, že každý další stav má o jednu jedničku navíc, dokud hodnota neobsahuje samé jedničky a poté se zase jedničky ze stejného směru ubírají.

```

StavKoderu: process (ABfiltr, predch_stav)
begin
    case predch_stav is
        when aretace =>
            if ABfiltr = "01" then
                akt_stav <= R1;
            elsif ABfiltr = "10" then
                akt_stav <= L1;
            else
                akt_stav <= aretace;
            end if;
        when R1 =>
            if ABfiltr = "11" then
                akt_stav <= aretace;
            elsif ABfiltr = "00" then
                akt_stav <= R2;
            else
                akt_stav <= R1;
            end if;
        when R2 =>
            if ABfiltr = "01" then
                akt_stav <= R1;
            elsif ABfiltr = "10" then
                akt_stav <= R3;
            else
                akt_stav <= R2;
            end if;
        when R3 =>
            if ABfiltr = "00" then
                akt_stav <= R2;
            elsif ABfiltr = "11" then
                akt_stav <= aretace;
            else
                akt_stav <= R3;
            end if;
        when L1 =>
            if ABfiltr = "11" then
                akt_stav <= aretace;
            elsif ABfiltr = "00" then
                akt_stav <= L2;
            else
                akt_stav <= L1;
            end if;
        when L2 =>
            if ABfiltr = "10" then
                akt_stav <= L1;
            elsif ABfiltr = "01" then
                akt_stav <= L3;
            else
                akt_stav <= L2;
            end if;
        when L3 =>
            if ABfiltr = "00" then
                akt_stav <= L2;
            elsif ABfiltr = "11" then
                akt_stav <= aretace;
            else
                akt_stav <= L3;
            end if;
        when others =>
            akt_stav <= aretace;
    end case;
end process;

```

Výpis 2: Popis stavového automatu v jazyku VHDL

* Advanced HDL Synthesis *	
Analyzing FSM <FSM_0> for best encoding. Optimizing FSM <predch_stav/FSM> on signal <predch_stav[1:7]> with one-hot encoding.	
State	Encoding
aretace	0000001
r1	0000010
r2	0001000
r3	0010000
l1	0000100
l2	0100000
l3	1000000

Výpis 3: Tabulka FSM se zakódovanými stavy

4.3 Ovládání čítače a asynchronní reset obvodu

Ovládání čítače a asynchronní reset je nadefinován v jednom procesu, který má v citlivostním seznamu uveden jako signál s nejvyšší prioritou signál CLR, který je aktivní v logické 1 a představuje asynchronní reset, při kterém je nastaven stavový automat do výchozího stavu „Aretace“ a vynulován obousměrný čítač kroků. Ovládání čítače je umožněno pouze při náběžné hraně hodinového signálu CLK, kdy se zjišťuje aktuální stav automatu. Při stavu Aretace a zároveň při předchozím stavu automatu v minulé náběžné hraně hodinového signálu CLK, byl-li ve stavu R3 nebo L3 se čítač buď inkrementuje, nebo dekrementuje.

```
Pocet_kroku_koderu: process (CLR, CLK, akt_stav)
variable stav_kroku : natural range 0 to 999 := 0;
begin
    if (CLR='1') then
        predch_stav <= aretace;
        stav_kroku := 0;
    elsif (CLK'event and CLK = '1') then
        if akt_stav = aretace then
            if predch_stav = R3 then
                if stav_kroku = 999 then stav_kroku := 0;
                else stav_kroku := stav_kroku + 1;
                end if;
            elsif predch_stav = L3 then
                if stav_kroku = 0 then stav_kroku := 999;
                else stav_kroku := stav_kroku - 1;
                end if;
            end if;
        end if;
        predch_stav <= akt_stav;
    end if;
end process;
```

Výpis 4: Popis asynchronního resetu a čítače v jazyku VHDL

4.4 Konfigurace a universálnost použití číslicového modulu

Konfigurace je poslední fáze po implementaci, ve které byl vytvořen konfigurační soubor s příponou .bit a z něho vygenerován soubor s příponou .mcs. Tento soubor byl pak uložen do paměti FLASH PROM na desce FPGA, aby i po odpojení napájení bylo možno obvod znovu použít. Vzhledem k tomu, že bylo použito ke konfiguraci číslicového modulu jen standardních prvků, je možno tento nakonfigurovaný obvod použít téměř pro všechny typy FPGA. Tento obvod je univerzální nejen pro FPGA ale i pro většinu typů rotačních kodérů až na pár výjimek. Tyto výjimky mají v aretované poloze při standardním zapojení výstupy v úrovni logické nuly. Pro tyto typy kodéru je zapotřebí pro použití tohoto číslicového modulu upravit konfiguraci obvodu ve VHDL kódu tím, že se oba vstupy po navzorkování znegují, pak už číslicový modul bude pracovat správně.

5 Experimentální ověření a zhodnocení dosažených výsledků

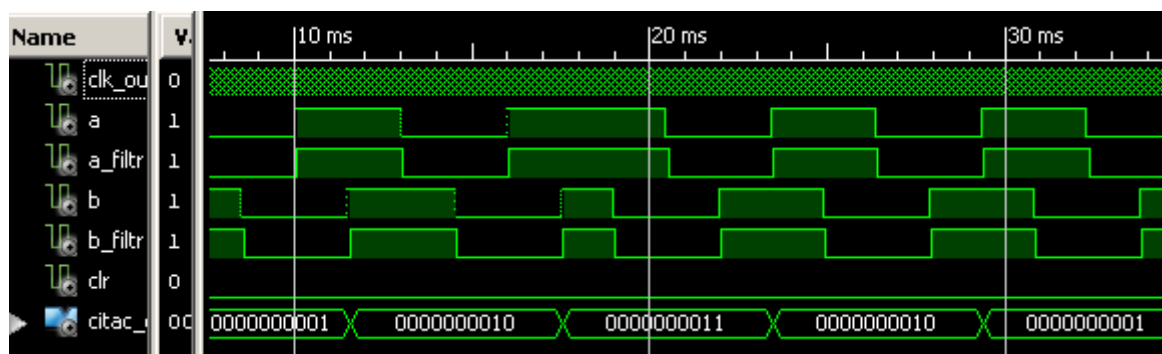
Ověření proběhlo jednak v simulaci v programu ISim, který je součástí programu ISE Design Suite, a také prakticky několika zátěžovými zkouškami zaměřenými na hlavní chyby inkrementálních kodérů. Pro ověření v simulaci byl vytvořen nejdříve nový zdrojový dokument typu VHDL Test Bench, v kterém se generují vstupní signály pro testovaný modul. Výstupem simulace jsou pak časové průběhy výstupních signálů modulu.

5.1 Ověření funkčnosti dekódování kvadrturních signálů v simulaci

V simulacích budou použity tyto signály:

clk_out	signál pro řízení vzorkování signálů A a B aktivní při náběžné hraně
a	výstupní signál A z rotačního kodéru
b	výstupní signál B z rotačního kodéru
a_filtr	vyfiltrovaný výstupní signál A z rotačního kodéru
b_filtr	vyfiltrovaný výstupní signál B z rotačního kodéru
clr	resetovací signál CLR
citac_out	stav obousměrného čítače kroků v binárním tvaru

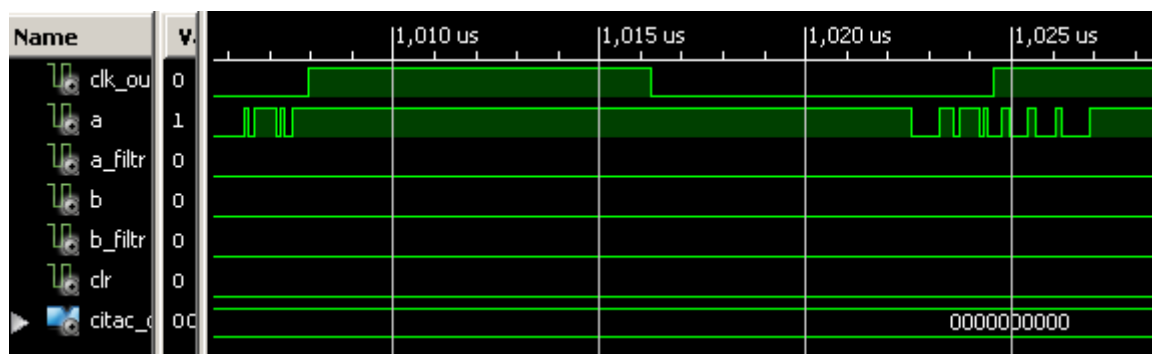
Při ověření funkčnosti byly nasimulovány kvadrturní signály o frekvenci 333 Hz, což odpovídá při 20 krocích na otáčku rychlosti otáčení kodéru 1000 ot/min. Průběh vstupních signálů byl nasimulován na tři kroky otáčení doprava a dva kroky doleva. Vzorkování signálů bylo provedeno frekvencí 30 kHz a na dvě platné hodnoty.



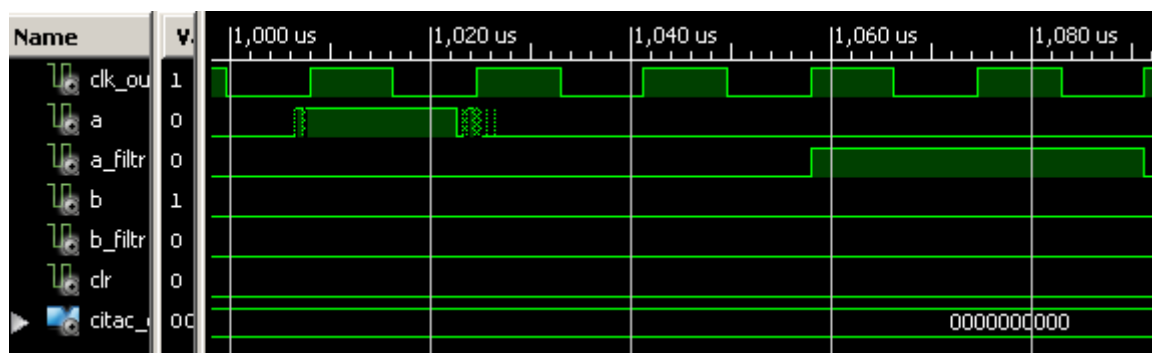
Obrázek 24: Simulace dekódování kvadrturních signálů

Z průběhu simulace je vidět v posledním signálu inkrementaci a dekrementaci stavu čítače na základě vstupních stavů signálů.

Další simulace byla zaměřena na filtraci zákmitů, kdy byly nasimulované impulzy o délce v rozmezí 100 ns až 15 μ s, které vycházely z reálných minimálních a maximálních naměřených hodnot při otáčení kodéru. Snímání signálů bylo provedeno frekvencí 30 kHz a na dvě platné hodnoty.



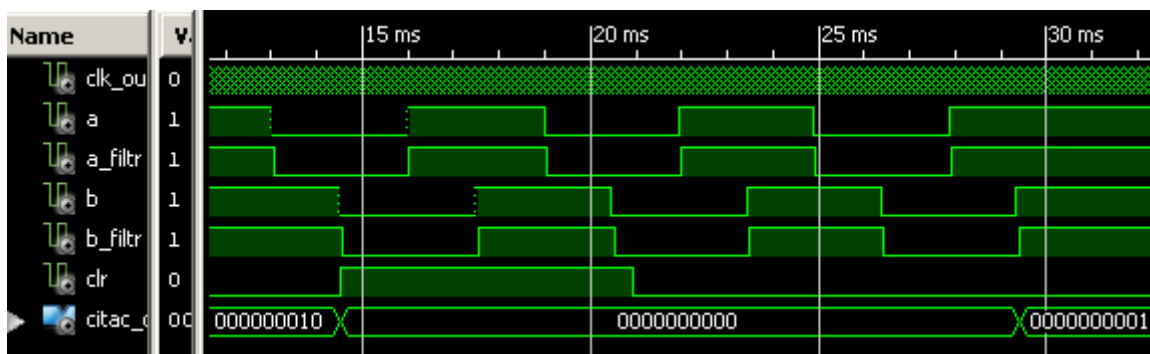
Obrázek 25: Simulace filtrování zákmitů



Obrázek 26: Simulace filtrování zákmitů

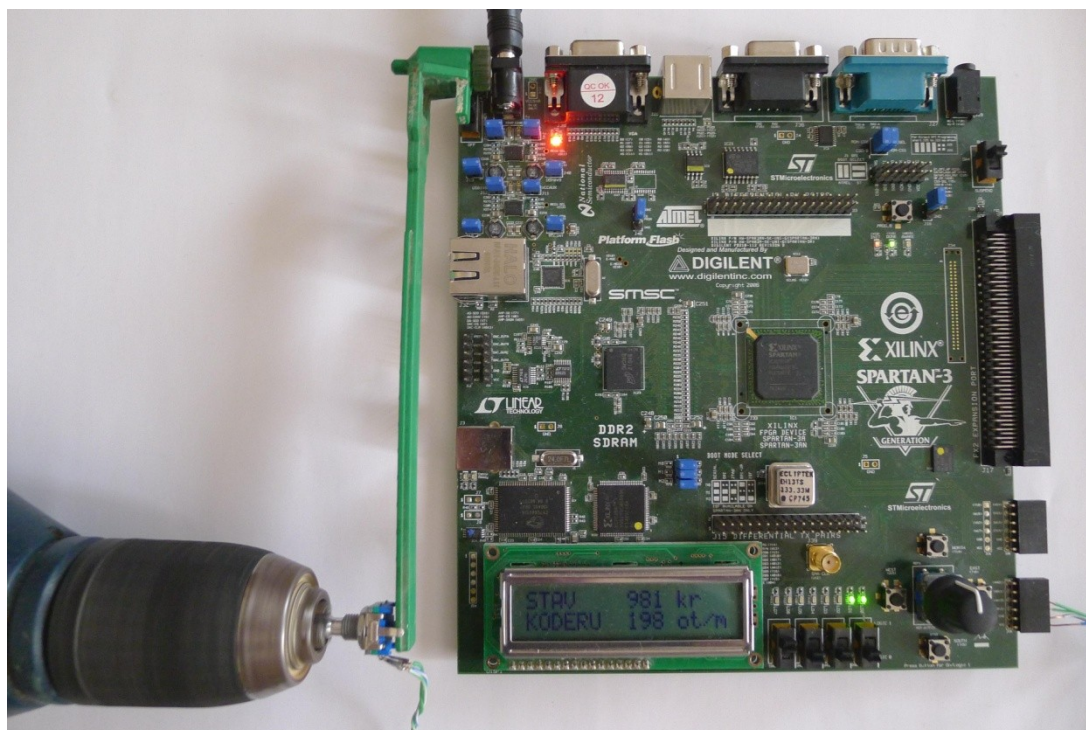
Z průběhu simulace na obrázku č. 25 je vidět, že při vzorkování a filtraci na dva platné signály je ještě velká pravděpodobnost, že může dojít k navzorkování dvou po sobě jdoucích hodnot o stejné logické hodnotě a vyhodnocení nechtěného impulsu. V simulaci na obrázku č. 26 je už vidět, že k navzorkování došlo tím, že na signálu a_filtr vznikl impuls. Dojde-li však v době zákmitů k navzorkování dvou hodnot po sobě jdoucích, nemělo-by dojít k chybnému dekódování, protože k zákmitům na vstupních signálech nedochází ve stejnou dobu (doba zákmitů vstupního signálu A se nepřekrývá se zákmity na signálu B), přesto je lepší zvýšit počet platných hodnot signálů pro filtraci, aby se snížila možnost nevhodného navzorkování výstupního signálu z kodéru. Druhá možnost je snížit frekvenci vzorkování tak, aby po dobu zákmitů nebylo možno dvakrát navzorkovat signál. To vede ale ke snížení maximálních otáček, které je obvod schopen dekódovat.

Poslední simulace je zaměřena na asynchronní reset pomocí signálu CLR, při kterém dojde k nastavení obvodu do výchozího stavu. Signál CLR je aktivní v logické jedničce, při které dojde k vynulování obousměrného čítače a nastavení stavového automatu do výchozího stavu Aretace. Dokud je signál v úrovni logické jedničky, nedochází k inkrementaci ani dekrementaci čítače.



Obrázek 27: Simulace asynchronního resetovacího signálu CLR

5.2 Praktické ověření funkčnosti reálného číslicového modulu

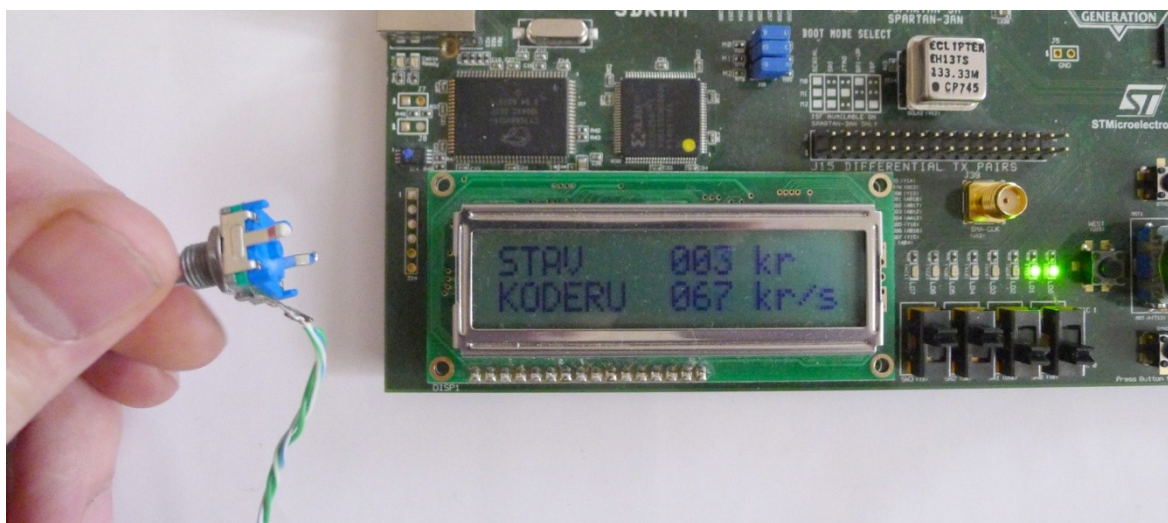


Obrázek 28: Časový test modulu

Praktické ověření proběhlo v několika testech, při kterých byly nastaveny tyto parametry obvodu: frekvence vzorkování 1 MHz a filtrace na tři platné hodnoty. Aby bylo možné jednoduše zjistit výsledky, byl obvod rozšířen o konfigurovatelné zobrazení hodnot na displeji FPGA, kde byl zobrazen počet kroků nebo otáček a rychlost otáčení v krocích za sekundu nebo v otáčkách za minutu.

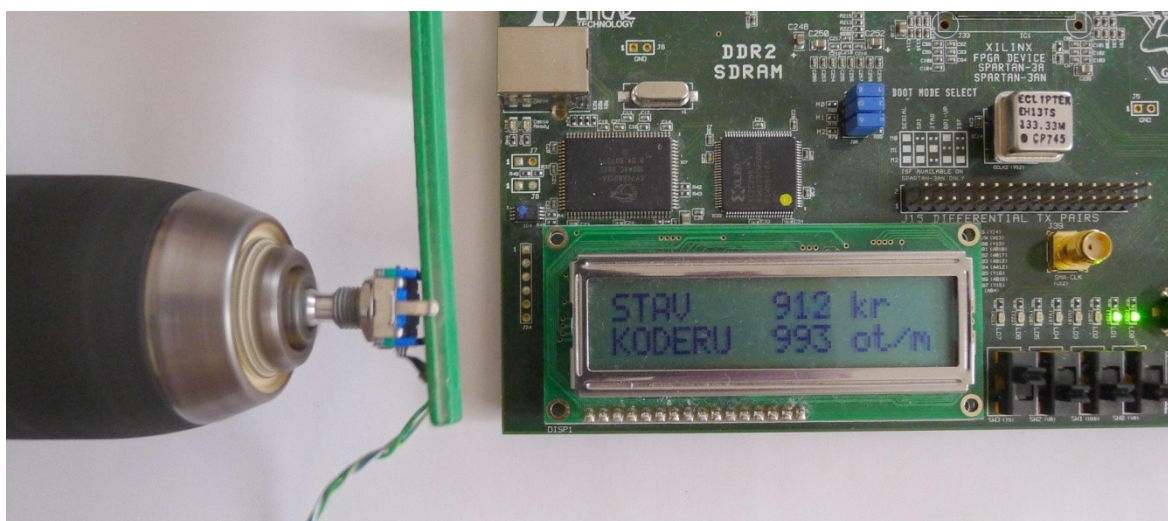
První test byl časový, při kterém byla poznačena výchozí poloha kodéru, čítač kroků byl vynulován a poté byl rotační kodér roztočen rychlostí cca 200 ot/s a ponechán v provozu 3 minuty na každý směr otáčení. Po vypnutí bylo ručně kodérem nastaveno na čítači kroků nejbližší číslo dělitelné 20, což je počet kroků na otáčku a bylo zjištěno, že kodér je ve své výchozí poloze čím se dokázalo bezchybné počítání kroků. Toto ověření proběhlo dvakrát a vždy s bezchybným výsledkem.

Druhý test, který se také opakoval dvakrát, byl zaměřen na vibrace a velmi malé otáčky, kdy se znovu poznačila výchozí poloha kodéru a rovněž čítač kroků byl vynulován a poté po dobu jedné minuty se ručně rychle měnil směr otáčení a po dobu další minuty se rotačním kodérem pootáčelo rychlostí cca 1 ot/min a také se měnil směr otáčení hlavně při nedokončeném kroku kodéru. Po skončení se opět pootočilo s kodérem tak ať je na čítači číslo dělitelné 20. Test dopadl rovněž s výborným výsledkem, protože se kodér nacházel ve své výchozí poloze.



Obrázek 29: Testování vibrací a velmi malých otáček

Třetí test byl zaměřen na maximální otáčky, které je možno pomocí kodéru P-RE20 snímat a při kterých již budou kvadratické signály natolik rušeny zákmity, že je nebude možno dekodovat. Nejdříve byl test proveden s akumulátorovou vrtačkou s maximální hodnotou 1000 ot/min, kdy byla naměřena hodnota 993 ot/min. Poté byl test proveden na soustruhu s maximální hodnotou 2200 ot/min, kdy bylo naměřeno 2187 ot/s. Poslední test byl proveden s ruční vysokootáčkovou bruskou s hodnotou 25000 ot/min, kdy se pomocí regulátoru měnily otáčky od cca 4000 ot/min až po maximum. Při otočení regulátoru na maximum ukazoval číslíkový modul hodnotu kolem 24580 ot/min, ale po asi sedmi vteřinách při maximálních otáčkách se rotační kodér tepelným třením kontaktů trvale poškodil a přestal plnit svou funkci. Maximální otáčky, při kterých by byly kvadratické signály natolik rušeny zákmity, se sice nepodařilo dosáhnout, ale tímto se dokázalo, že při vyšších otáčkách se zkrátily zákmity natolik, že jejich doba trvání mohla být maximálně 57 μ s. Vyšší otáčky již nemá smysl zkoušet kvůli konstrukčním vlastnostem kodéru, který je konstruován na ruční ovládání a jeho životnost je uváděna na 15 000 otočení, i přesto vydržel nad míru jeho udávaným konstrukčním vlastnostem a životností.



Obrázek 30: Testování maximálních otáček

6 Závěr

Vytvoření číslicového modulu pro čtení inkrementálního rotačního kodéru v programovatelném hradlovém poli byla pro mě výzva, protože v této oblasti jsem neměl dosud žádné zkušenosti jak s jazykem VHDL tak už vůbec ne se samotnými FPGA. Psaní v jazyku VHDL mi zpočátku dělalo problém zejména specifické vlastnosti tohoto jazyka a dodržováním zásad pro pozdější syntézu do reálného obvodu. Jsem rád, že se mi podařilo zvládnout toto odvětví elektrotechniky, na které jsem dostal nový náhled a spatřuji v něm velký potenciál v mém studovaném oboru měřicí a řídicí techniky, zvláště pro svůj výkon, který spočívá v paralelismu při zpracování vstupních signálů a ve variabilitě a možnosti poměrně snadných úprav vytvořených obvodů v FPGA. Číslicový obvod vytvořený v FPGA se mi podařilo úspěšně zrealizovat a testováním dokázat jeho spolehlivost i v náročných podmínkách. Navíc jsem si vyzkoušel ovládání periférií, jako byl displej osazený na desce FPGA, na kterém jsem zobrazil všechny změřené i vypočtené hodnoty. Simulacemi a měřením jsem zjistil, že je pro správné dekodování kvadrurních signálů z mechanických kodérů, důležité určit minimální počet po sobě navzorkovaných logických hodnot, který by měl obsahovat alespoň 3 vzorky pro určení platné hodnoty signálů s vhodnou frekvencí jejich vzorkování.

7 Literatura

1. ŠTASTNÝ, Jakub. *FPGA prakticky*. 1. vyd. Praha: BEN, 2010. 200 s. ISBN 978-80-7300-261-9.
2. ŠTASTNÝ, Jakub.: Programovatelná hradlová pole. *Automatizace*, 2008, roč. 51, č. 1, s. 9–14. ISSN 0005-125X.
3. KOLOUCH, Jaromír.: Programovatelná logické obvody a hradlová pole – moderní stavební prvky číslicových systémů. *Automatizace*, 2009, roč. 52, č. 1, s. 36–39. ISSN 0005-125X.
4. KOLOUCH, Jaromír. *Programovatelné logické obvody: příklady konstrukcí a poznámky k práci s jazykem VHDL: počítačové cvičení*. 1. vyd. Brno: Vysoké učení technické v Brně, 2001. 56 s. ISBN 80-214-1966-0.
5. KOLOUCH, Jaromír. Programovatelné logické obvody a modelování číslicových systémů v jazycích ABEL a VHDL: přednášky - počítačové cvičení. 1. vyd. Brno: Vysoké učení technické v Brně, 2000. 83 s. ISBN 80-214-1733-1.
6. NOVÁK, Petr. Rotační inkrementální senzory. *Automa*, 2002, roč. 8, č. 10, s. 32–33. ISSN 1210-9592.
7. PINKER, Jiří - POUPA, Martin. *Číslicové systémy a jazyk VHDL*. 1. vyd. Praha: BEN, 2006. 352 s. ISBN 80-7300-198-5.
8. EGERT, Jan. *Algoritmy pro kódování stavů pro konečné automaty*. Praha: České vysoké učení technické v Praze. Fakulta elektrotechnická, 2007. 46 s. Vedoucí bakalářské práce Ing. Petr Fišer.
9. KAŠÍK, Vladimír. *Kurs kybernetiky pro SŠ: Programovatelná hradlová pole FPGA: učební text a návody do cvičení*. [elektronická skripta]. 1. vyd. Ostrava: 2012. 59s
10. *Spartan-II FPGA Family Data Sheet*. [online]. Xilinx, Inc., 2008. DS001 (v2.8) 13.6.2008.[cit. 2013-3-19]. URL <http://www.xilinx.com/support/documentation/data_sheets/ds001.pdf>
11. *Spartan-3 Generation FPGA User Guide*. [online]. Xilinx, Inc., 2008. UG331 (v1.8) 13.6.2011.[cit. 2012-7-21]. URL <http://www.xilinx.com/support/documentation/user_guides/ug331.pdf>
12. XILINX. 7.2 *Xilinx LCA*. [online]. c1997, 15.9.2004.[cit.2013-1-11]. URL <<http://www.wdsu.uqac.ca/~daudet/Cours/Vlsi/DOCUMENTS/repertoire435/Cours-MJS-Smith/CH07/CH07.7.htm>>
13. *Princip optických enkodérů polohy pro řízení motorů* [online]. 9.6.2006.[cit. 2013-2-15]. URL <http://jonatan.spse.pilsedu.cz/~mazanec/opticke_ekodery.htm>

14. MONTY. *AS5304 / AS5306 - lineární magnetické kodéry s inkrementálním výstupem a rozlišením 25 μ m / 15 μ m od austriamicrosystems*. 22.8.2010.[cit. 2013-2-15]. URL <<http://www.spezial.cz/news/linearni-magneticke-enkodery-s-inkrementalnim-vystupem.html>>
15. ELTRA. *I25-01-398-001-EH-0412.pdf*. 9.12.2012.[cit. 2013-1-12]. URL <<http://www.pewatron.com/en/datasheets/sensors>>
16. KOPELANT Jiří. *P-RE20*. 30.7.2005.[cit. 2011-8-7]. URL <<http://www.gme.cz/dokumentace/532/532-085/dsh.532-085.1.pdf>>
17. Technické listy kodérů. [online]. [cit. 2012-12-20]. URL <<http://www.pewatron.com>>
18. Technické listy kodérů. [online]. [cit. 2012-12-29]. URL <<http://www.encoder.com>>
19. Technické listy kodérů. [online]. [cit. 2012-12-20]. URL <<http://www.oemautomatic.cz>>
20. Technické listy kodérů. [online]. [cit. 2012-12-19]. URL <<http://www.eu.omron.com/ema>>
21. Technické listy kodérů. [online]. [cit. 2012-12-18]. URL <<http://www.automationdirect.com>>
22. Technické listy kodérů. [online]. [cit. 2012-12-20]. URL <<http://www.lenord.de>>
23. Technické listy kodérů. [online]. [cit. 2012-12-18]. URL <<http://www.agilent.com/semiconductors>>
24. Technické listy kodérů. [online]. [cit. 2012-12-19]. URL <<http://www.sikoproducts.com>>
25. Technické listy kodérů. [online]. [cit. 2012-12-20]. URL <<http://www.am.pepperl-fuchs.com>>
26. Technické listy kodérů. [online]. [cit. 2012-12-19]. URL <<http://www.kubler.cz>>
27. Technické listy kodérů. [online]. [cit. 2012-12-18]. URL <<http://www.leinelinde.se>>
28. Technické listy kodérů. [online]. [cit. 2012-12-20]. URL <<http://www.givimisure.it>>
29. Technické listy kodérů. [online]. [cit. 2012-12-19]. URL <<http://www.megatron.de>>
30. Technické listy kodérů. [online]. [cit. 2012-12-20]. URL <<http://www.baumer.com>>

8 Přílohy

- I. Technické listy kodérů
- II. Kompletní projekt Číslicového modulu v programu ISE Design Suite
- III. VHDL kód Číslicového modulu pro čtení rotačního kodéru